NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN INTERPOLATION APPROACH TO OPTIMAL TRAJECTORY PLANNING FOR HELICOPTER UNMANNED AERIAL VEHICLES

by

Jerrod C. Adams

June 2012

Thesis Advisor: Wei Kang
Second Reader: Hong Zhou

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2012	Master's Thesis

4. TITLE AND SUBTITLE **An Interpolation Approach to Optimal Trajectory Planning for** Helicopter Unmanned Aerial Vehicles	5. FUNDING NUMBERS
6. AUTHOR(S) Jerrod C. Adams	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited	A

13. ABSTRACT (maximum 200 words)

This thesis explores numerical methods to provide real-time control inputs to achieve an optimal trajectory which minimizes the time required for a Helicopter Unmanned Aerial Vehicle (HUAV) to reorient to a given target. A library of optimal trajectories is populated using a pseudospectral computational algorithm applied to the mathematical model developed by the National University of Singapore and Singapore Department of Defense to simulate flight characteristics for their HeLion small scale HUAV system. The model is a complex system of non-linear differential equations—fifteen state variables and four control variables—used to simulate the aerodynamic forces on the HUAV. Then, using the library of optimal trajectories for known target locations, we apply interpolation methods to provide control inputs in order to intercept an attack heading to a target more quickly than an online, full scale optimization approach. All simulations in this thesis are modeled using the MATLAB program.

14. SUBJECT TERMS Nonlinear Model, State and Control Variables, Cost Function, , Trajectory Optimization, Target Heading Intercept, Bilinear Interpolation	15. NUMBER OF PAGES 65
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

AN INTERPOLATION APPROACH TO OPTIMAL TRAJECTORY PLANNING FOR HELICOPTER UNMANNED AERIAL VEHICLES

Jerrod C. Adams
Major, United States Army
B.S., United States Military Academy, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
June 2012

Author: Jerrod Adams

Approved by: Wei Kang
Thesis Advisor

Hong Zhou
Second Reader

Carlos Borges
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis explores numerical methods to provide real-time control inputs to achieve an optimal trajectory which minimizes the time required for a Helicopter Unmanned Aerial Vehicle (HUAV) to reorient to a given target. A library of optimal trajectories is populated using a pseudospectral computational algorithm applied to the mathematical model developed by the National University of Singapore and Singapore Department of Defense to simulate flight characteristics for their HeLion small scale HUAV system. The model is a complex system of non-linear differential equations—fifteen state variables and four control variables—used to simulate the aerodynamic forces on the HUAV. Then, using the library of optimal trajectories for known target locations, we apply interpolation methods to provide control inputs in order to intercept an attack heading to a target more quickly than an online, full scale optimization approach. All simulations in this thesis are modeled using the MATLAB program.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. LITERATURE REVIEW 1
A. BACKGROUND 1
1. Helicopter Unmanned Aerial Vehicle Applications 1
2. Trajectory Optimization 2
a. Database Interpolation 3
b. Online Solvers 4
B. PSEUDOSPECTRAL ALGORITHM EXISTING RESULTS 5
1. Previous Work 5
2. Current Work 6
C. CONCLUSION 6

II. PROBLEM FORMULATION 9
A. HUAV MODEL 9
B. OPTIMAL CONTROL PROBLEM DEFINITION 13
1. Cost 13
2. State and Control Bounds 14
C. CONSTRAINTS 15

III. SOLUTION APPROACH 17
A. BUILDING THE LIBRARY 17
1. Finding Optimal Trajectories 17
2. Pseudospectral Method 18
B. INTERPOLATION 19
1. One Dimension 19
2. Two Dimensions 21

IV. RESULTS 25
A. LIBRARY OF OPTIMAL TRAJECTORIES 25
B. INTERPOLATION 26
1. One Dimension Interpolation 26
2. Two Dimension Interpolation 30

V. CONCLUSIONS AND FUTURE WORK 41
A. CONCLUSIONS 41
B. RECOMMENDATIONS FOR FUTURE RESEARCH 42
1. Incorporate Interpolation Method in Three Dimensions 42
2. Further Investigation into Variable Mesh 42
3. Code for an Onboard Platform 43
4. Incorporate a Closed-Loop Feedback Control System 43

LIST OF REFERENCES 45

INITIAL DISTRIBUTION LIST 47

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. Conceptual Image of an AH-6U HUAV (From: [2])....1
Figure 2. HeLion, a small-scale HUAV helicopter (From [16])....9
Figure 3. Variables and Frames of Reference (from [16])...11
Figure 4. Target Grid, Optimal Trajectories....17
Figure 5. Optimal trajectory, target at [20,5,-2] NED frame....19
Figure 6. Two Dimension Bi-Linear Interpolation Grid....22
Figure 7. 1D Interpolated trajectory, [20 5-2]East axis view....27
Figure 8. One dimension interpolated trajectory for a target located at [20 5-2]North axis view....28
Figure 9. Final HUAV heading to target angular error, one dimension interpolation in East axis....29
Figure 10. Final HUAV heading to target angular error, one dimension interpolation in North axis....30
Figure 11. Final angular error between the optimal and actual heading at the interpolated trajectory's final state....32
Figure 12. Final angular error with additional 5 meter grid from North∈{0 to 10 }and East∈{0 to 10} ...34
Figure 13. Final angular error with additional 5 meter grid from North∈{10 to 20 }and East∈{0 to 10} ..35
Figure 14. Final angular error with additional 5 meter grid from North∈{10 to 30 }and East∈{0 to 20} ..35
Figure 15. Final angular error with additional 5 meter grid from North∈{0 to 50 }and East∈{0 to 20} ...36

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Physical Meanings of the State Variables (From [14]).....10
Table 2. Physical meaning of the control variables (From [14]).....11
Table 3. Bounds for the State and Control Variables (After [14]).....15
Table 4. Library of optimal trajectories.....26
Table 5. Two-dimension interpolation final heading to target angular error, using the 10 meter library.....31
Table 6. Final angular errors using the mixed mesh Library.....37
Table 7. Summarization of pseudospectral vs interpolation results for test targets.....38
Table 8. Pseudospectral vs. 2-D interpolation method results over *North, East* ∈{5, 10, …45} 39

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

HUAV	Helicopter Unmanned Aerial Vehicle
UA	Unmanned Aircraft
ADL	Armament Data Line
DOF	Degree of Freedom
PS	Pseudospectral
LGL	Legendre-Gauss-Lobatto quadrature nodes
ODE	Ordinary Differential Equation

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to offer my sincerest appreciation to the entire faculty in the Math department at the Naval Postgraduate School for their instruction and assistance during my graduate studies. I would especially like to thank: Les Carr for his help in developing the animation code that helped me visualize the problem in this thesis; Frank Giraldo for his help in understanding large scale numerical methods in MATLAB; Carlos Borges for his leadership, professionalism, and collaborative work atmosphere he fostered within the department; Bard Mansager for his help during the admission process and as an advisor during my time at NPS; Hong Zhou for her input and suggestions to make this paper stronger; and most importantly my advisor Wei Kang for his patience, wisdom, and freedom that allowed me to grow and develop as a mathematician during this process. Finally I would like thank my wife and children for their support as I spent long hours at school and at home working on this problem.

THIS PAGE INTENTIONALLY LEFT BLANK

I. LITERATURE REVIEW

A. BACKGROUND

1. Helicopter Unmanned Aerial Vehicle Applications

Helicopter Unmanned Aerial Vehicles (HUAVs) conduct various military missions, from surveillance to search and rescue. They serve as a unique tool for the commander which broadens situational awareness on the battlefield. The ability to see, target, and destroy the enemy without putting soldiers in danger makes HUAVs an emerging lethal and non-lethal weapon of choice that will continue to transform how the Army prosecutes future operations and ultimately save lives. In a 2010 roadmap for Unmanned Aerial Systems [1], General Dempsey, the Army's Training and Doctrine Commanding General, noted that "in the future, all UA groups will possess lethal weapons capabilities" like the image in Figure 1 of the AH-6U unmanned scout attack helicopter being developed by Boeing.

Figure 1. Conceptual Image of an AH-6U HUAV (From: [2])

This paper explores finding optimal control inputs for the HUAV sensor pointing and armed attacks. In both scenarios, target information is passed to the HUAV and it must rapidly re-orient from its initial position to some final position in order to either put the target within the sensor's field of regard or employ its weapons payload. The specific example presented here involves our HUAV flying straight and level and then reorienting to a target in minimum time. This particular scenario was inspired by another NPS thesis, which looked at arming small and lightweight UAV's for counter-sniper operations [3]. The UAV must line up a direct heading to the target and fire its weapon system to either destroy the target or mark the area or personnel with an invisible chemical that is only traceable by special equipment carried by friendly forces.

2. Trajectory Optimization

In order to re-orient a HUAV from an initial position to a final firing position, a path is required. Once a feasible path is computed, control inputs are required to maneuver the aircraft along that path and tracking feedback control is also required, a separate issue not addressed in this thesis. Trajectory planning determines all states and control inputs as functions of time for the required path. To accomplish the complete trajectory subject to a performance measure, or cost, such as minimal time, an optimal trajectory is needed.

In the case of this paper, an optimal trajectory is defined as a complete set of states and controls to maneuver the HUAV in as little time as possible, while constrained to the admissible flight envelope of the

airframe. There are two general ways of achieving optimal trajectories in real time, online optimal trajectory solvers or searching and interpolating over a database [4]. Benefits and limitation of both methods are presented first.

a. Database Interpolation

In [5] and [6], Atkeson searches a large trajectory database and interpolates to generate real-time optimal trajectories for a pendulum and humanoid robot respectively. He claims that online solvers are computationally expensive and may not be feasible. His database, or library, consists of a grid of discretized state cells and assigns a control action to each cell. To find a global optimal trajectory, many local trajectories are pieced together.

The problem needed to overcome is to make each piecewise trajectory consistent with the following. A trajectory can be made consistent with a neighbor by using the neighboring trajectory as an initial trajectory in the local optimization process, or by using the value function from the neighboring trajectory to generate the initial trajectory in the local optimization process [5]. Each grid element stores the trajectory that starts at that point and achieves the lowest cost.

Lastly, Atkeson proposed a varying step size trajectory library on an adaptive grid of initial conditions in order to strike a balance between library size and performance. In a uniform grid, if the step size is too small, the performance of the optimal trajectory increases but the library becomes too large. Likewise, if

the step size is too large, memory savings in the library are realized, but solutions may perform poorly or not converge. In order to store trajectories on an adaptive grid of initial conditions, optimal trajectories are generated and then stored into a library incrementally based on performance [6].

b. Online Solvers

In [7], researchers at the California Institute of technology proposed a method of finding real-time nonlinear trajectories for unmanned aerial vehicles from a start point to a final destination point to maximize the probability of not-being detected function in minimal total flight time. They claim that searching a trajectory database and piecing together trajectories can be too time consuming and sub-optimal for large dimension problems.

Their technique uses splines as base functions for parameterization (used for their flexibility and calculating their derivatives as discussed later), and a software package called Nonlinear Trajectory Generation (NTG) designed at Caltech by Mark B. Milam et al. [8] to solve optimal control problems in real-time using the online solver method. A spline is a collection of piecewise, low order polynomials patched together in such away so that the resulting trajectory has several continuous derivatives at all points. In [7], Murray claims that splines are ideal for optimal control problems because each segment of the spline's piecewise polynomials approximate the trajectory and satisfies the constraints locally. This methodology is based on finding trajectory curves in a lower dimensional space, then parameterizing

the curves with B-splines, and finally using sequential quadratic programming to satisfy the optimization objectives as well as the constraints.

The use of splines is important to this online solver approach. Researchers have used cubic splines since as early as 1973 to solve optimal control problems. In [9], Neuman and Sen used cubic spline collocation on a uniform mesh to solve several optimal control problems. Since then, splines have been used extensively to formulate and solve optimal control problems numerically [7]. Additionally, spline functions have been extensively used as an approximation tool in areas such as curve fitting, motion planning and trajectory generation for mechanical systems [10]. However, the NTG online solver mentioned in Murray's work is applicable to flat systems only, which is not the case for the HUAV operating in three-dimensional space.

B. PSEUDOSPECTRAL ALGORITHM EXISTING RESULTS

1. Previous Work

In [11], Major Gatzke used a pseudospectral method to achieve straight line flight optimal trajectories through the use of an older version of the HeLion HUAV mathematical model. The pseudospectral method he utilized is a discretization method developed by a group of researchers from NPS [12] and [13] to achieve optimal trajectories which meet a performance cost function given initial and final state requirements, without exceeding constraints. The cost function optimized in Gatzke's work was minimal time while attempting to avoid various obstacles, but he was never able to include the obstacles into his work. He was able to achieve minimal time trajectories for straight

line flight up to ten meters using the pseudospectral method, but he concluded that more work was needed to reduce the error induced when the pseudospectral method calculated the optimal path by integrating the non-linear HUAV model

2. Current Work

Since MAJ Gatzke's work, an updated and improved HeLion mathematical model with fifteen state variables has produced much more feasible and stable results. There is simultaneous work being conducted now by Professor Kang and his colleagues [14] and [15], which has led to much more stable and feasible results, with the inclusion of the obstacles along the path. One interesting finding of their current work is bifurcation points in the optimal trajectories produced when minimizing time for the HUAV to fly around obstacles. These bifurcation points will be addressed in this paper as well.

C. CONCLUSION

This paper attempts to utilize portions of both the online solver and search and interpolate methods of finding optimal trajectories in real time. A database of optimal trajectories is created using the pseudospectral optimal control method for targets at various locations. Then we use bilinear two dimensional interpolations between four nearest neighbors in the database to interpolate the control inputs required to provide an approximate optimal trajectory.

The process of using interpolations from known optimal trajectories produces approximate but quickly computed near

optimal trajectories. Although not guaranteed to be optimal, the resulting trajectory should come close to a minimal cost, and reorient the HUAV to a direct heading to the target within a pre-determined threshold.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PROBLEM FORMULATION

A. HUAV MODEL

The mathematical model used to simulate the HUAV flight is a comprehensive nonlinear aerodynamic MATLAB model designed by researchers at the National University of Singapore [16] in conjunction with the Singapore Department of Defense to simulate the flight characteristics of their small scale HUAV "HeLion" seen in Figure 2. The model consists of a system of fifteen nonlinear differential equations governed by fifteen state variables and four control inputs, which can be widely adapted to other small-scale HUAV systems. [17].

Figure 2. HeLion, a small-scale HUAV helicopter (From [16])

The state variables are referenced in two different coordinate systems, the North-East-Down frame and the Body frame. The North-East-Down (NED) frame can be visualized if an aircraft were to fly over the surface of the earth, where North is along the usual positive Cartesian x-axis always towards the north pole, East is along the positive y-axis along the equator, and positive Down values are equitable to the contemporary negative z-axis, or

increasing in value towards the center of the earth. The body frame would always be in reference to the aircraft no matter what NED direction it is flying, straight ahead, to the aircraft's right, or down towards the bottom of the aircraft. The state variables are: the three dimensional coordinates of the helicopter's position in the NED frame; its angular orientation in roll, pitch, and yaw in the body frame; velocity along the x, y, and z axes of the body frame, angular rates in the body frame; flapping angles and the tail rotor gyro intermediate state. These state variables are grouped in the state vector below, with their associated variables and units in Table 1.

$$x = \left[p_x \;\; p_y \;\; p_z \;\; u \; v \; w \;\;\; \varphi \;\; \theta \;\; \psi \;\;\;\; p \; q \; r \;\;\; a_s \; b_s \; \delta_{ped,int} \right]^T$$

Table 1. Physical Meanings of the State Variables (From [14])

Variable	**Physical meaning**	**(unit)**
p_x, p_y, p_z	Position vector in NED-frame	(m)
u, v, w	Velocity vector in body-frame	(m/s)
φ, θ, ψ	Roll, pitch, and yaw angles in NED-frame	(rad/s)
p, q, r	Roll, pitch, and yaw angular rate in body-frame	(rad/s)
a_s, b_s	Longitudinal and lateral tip-path-plane (TPP)	flapping angle
$\delta_{ped,int}$	Intermediate state in yaw rate gyro dynamics	

The control input variables govern lateral and longitudinal cyclic inputs, collective input, and tail rotor torque inputs, grouped in the control vector below, with the variables and their physical meanings listed in Table 2.

$$u = \left[\delta_{lat} \quad \delta_{long} \quad \delta_{col} \quad \delta_{ped} \right]^T$$

Table 2. Physical meaning of the control variables (From [14])

δ_{lat}	Normalized aileron servo input
δ_{lon}	Normalized elevator servo input
δ_{col}	Normalized collective pitch servo input
δ_{ped}	Normalized rudder servo input

Figure 3 gives a spatial representation of the state and control variables as well as the body and NED reference frames.

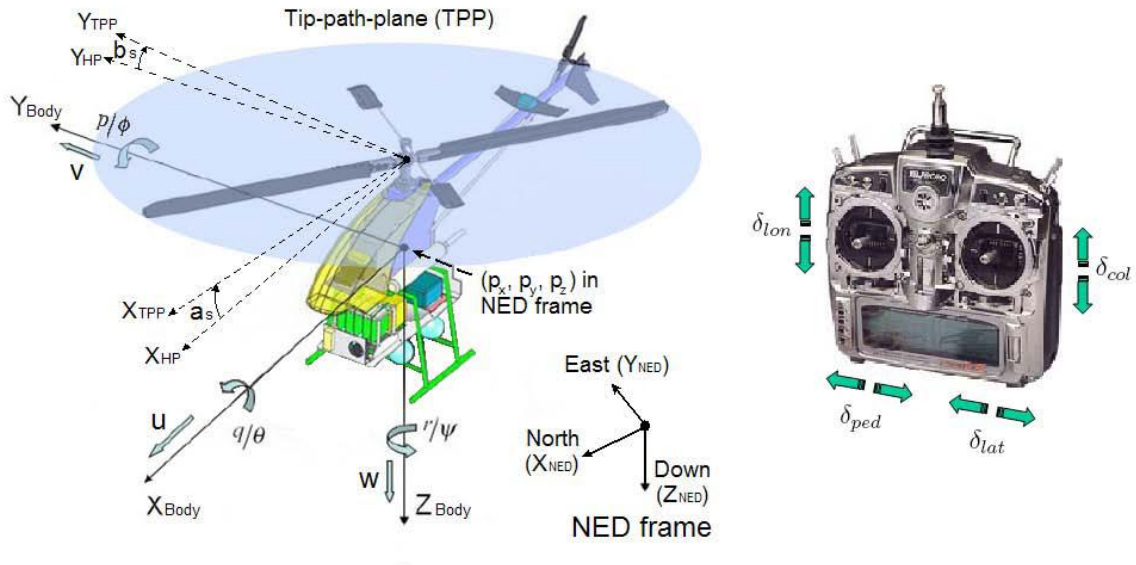


Figure 3. Variables and Frames of Reference (from [16])

For a full description of the highly technical rigid-body and kinematic equations used to model the HUAV's flight, the reader is referred to [16]. These non-linear differential equations can be briefly summarized into two sets of kinematic equations used to compare the relative

motions between the two coordinate frames and the six degree of freedom (DOF) rigid-body dynamics of the helicopter airframe.

The first kinematic equation relates translational motion and is given by

$$\dot{P}_n = B_{Body} * V_b$$

where $P_n = (p_x \; p_y \; p_z)^T$ is the position vector in the NED frame, $V_b = (u \; v \; w)^T$ is the velocity vector in the body frame, and B_{Body} is the transformation matrix that relates the HUAV's body frame angular headings to the NED frame and is given by

$$B_{body} = \begin{pmatrix} \cos\theta\cos\psi & \sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi & \cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi \\ \cos\theta\sin\psi & \sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi & \cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi \\ -\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta \end{pmatrix}$$

The second kinematic equation is for the rotational motion and is given as follows

$$\dot{\Omega}_n = S_{Body} * \Omega_b,$$

where $\Omega_n = (\phi \; \theta \; \psi)^T$ is the Euler angle vector, $\Omega_b = (p \; q \; r)^T$ is the angular rate vector in the body frame, and S_{Body} is the corresponding transformation matrix given by

$$S_{body} = \begin{pmatrix} 1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi / \cos\theta & \cos\varphi / \cos\theta \end{pmatrix}$$

The six DOF rigid-body dynamics of the HUAV airframe is represented by the following Newton-Euler equations

$$\dot{V}_b = -\Omega_b \times V_b + \frac{F_b}{m} + \frac{F_g}{m}$$
$$\dot{\Omega}_b = I^{-1}(M_b - \Omega_b \times I \cdot \Omega_b)$$

where m is the mass of the HUAV, $I = diag\{I_{xx}, I_{yy}, I_{zz}\}$ is the moment of inertia, F_g is the gravity force vector, F_b is the aerodynamic force vector and M_b is the aerodynamic moment vector given by

$$F_g = \begin{bmatrix} m \cdot g \cdot \sin\theta \\ m \cdot g \cdot \sin\varphi\cos\theta \\ m \cdot g \cdot \cos\varphi\cos\theta \end{bmatrix}$$

$$F_b = \begin{bmatrix} F_{bx} \\ F_{by} \\ F_{bz} \end{bmatrix} = \begin{bmatrix} X_{mr} + X_{fus} \\ Y_{mr} + Y_{fus} + Y_{tr} + Y_{vf} \\ Z_{mr} + Z_{fus} + Z_{hf} \end{bmatrix}$$

$$M_b = \begin{bmatrix} M_{bx} \\ M_{by} \\ M_{bz} \end{bmatrix} = \begin{bmatrix} L_{mr} + L_{vf} + L_{tr} \\ M_{mr} + M_{hf} \\ N_{mr} + N_{vf} + N_{tr} \end{bmatrix}$$

where the X, Y, Z, L, M, and N forces or moments represent much more complicated and expanded systems of equations dealing with the principle components of the HUAV including the main rotor, fuselage, tail rotor, vertical fin, and horizontal fin that will we will not cover here.

B. OPTIMAL CONTROL PROBLEM DEFINITION

1. Cost

To compute the minimum time, J, required for the HUAV to reorient to a target, the following problem of optimal control is utilized

$$\min J = \int_{t_0}^{t_f} 1 dt$$

subject to

$$\dot{x} = f(x,\ u)$$

$$x_{\min} \le x \le x_{\max},\ \ u_{\min} \le u \le u_{\max},\ \ z \le -5$$

$$x(t_0) = x_0,\ x(t_f) = c(x_f),\ t_f \text{ is unspecified}$$

Where $f(x,u)$ represents the HeLion model and $z \le -5$ is the minimum flight altitude allowed in the NED frame. The x and u limits are discussed along with the other bounds in the constraints section. Six of the state variables are prescribed at the initial time (p_x=0, p_y=0, p_z=-10, u=6 m/s, v=0, w=0), while the remaining states and final time are left free for the algorithm to optimize so long as the constraint function $c(x_f)$ is satisfied and t_f is minimal.

2. State and Control Bounds

The range searched for optimal trajectory solutions in NED-frame is from −50 to 50 meters for the North and East coordinates and −50 to -5 in the Down coordinate. The maximum of -5 in the Down direction is to prevent the HUAV from impacting the ground. The range (−1.5, 1.5 radians) for pitch and roll in the body-frame is implemented to prevent the helicopter from exceeding operational limits, and 2π radians for yaw to give complete freedom of direction. In the body-frame, the x velocity is bound from −10 to 10 m/s , the y and z velocities from −5 to 5 m/s, and the rotational velocity from −1.0 to 1.0 rad/s. The allowable ranges found in the manual of the helicopter model are used for the controls. These upper and lower bounds are summarized in Table 3.

Table 3. Bounds for the State and Control Variables (After [14])

p_x: (-50,50)	p_y: (-50,50)	p_z: (-50,-5)
u: (-10,10)	v: (-5,5)	w: (-5,5)
φ: (-1.5,1.5)	θ: (-1.5,1.5)	ψ:(-3.14,3.14)
p: (-1,1)	q: (-1,1)	r: (-1,1)
a_s:(-0.02,0.02)	b_s: (-0.02,0.02)	$\delta_{ped,\text{int}}$:(-1,1)
δ_{lat}: (-1,1)	δ_{lon}: (-1,1)	
δ_{col}: (-1,0)	δ_{ped}: (-1,1)	

The initial state condition of the HUAV in all scenarios is from straight and level flight at 6 m/s in the x direction, zero m/s in the y and z axes, while at an altitude of -10 meters and North and East position of zero meters each. All other initial states and controls were allowed to remain within the admissible bounds in order to meet the straight and level fight profile. The performance measure or cost used is the time to reorient from the initial state to the final state which met the end state criteria bounds.

C. CONSTRAINTS

The final state criteria that drives the optimal trajectory, $c(x_f)$, is the goal of the final heading of the HUAV, pointing directly at the target presented. In each scenario, the armament data line (ADL), an imaginary vector directed straight out of the nose of the HUAV in the x direction, is used to represent the aim of the sensor platform or weapon system. In order to bring the munitions or sensor into firing constraints, the HUAV needs to reorient from its initial heading to a heading directly pointed at the target. This optimal heading, $\Omega_{AngleofAttack}$, is measured as the difference between the vector from the

HUAV's final position to the target, Ω_{TGT}, and the HUAV's final heading, Ω_{UAV}, both in the NED frame as

$$\Omega_{AngleofAttack} = \Omega_{TGT} - \Omega_{UAV}$$

These vectors were normalized and represented as three dimensional angles measured in the roll, pitch, yaw states.

$$\Omega_{\mathrm{TGT}} = \frac{[p_{\mathrm{x}}\ p_{\mathrm{y}}\ p_{\mathrm{z}}]_{\mathrm{TGT}} - [p_{\mathrm{x}}\ p_{\mathrm{y}}\ p_{\mathrm{z}}]_{\mathrm{UAV}}}{\left\| [p_{\mathrm{x}}\ p_{\mathrm{y}}\ p_{\mathrm{z}}]_{\mathrm{TGT}} - [p_{\mathrm{x}}\ p_{\mathrm{y}}\ p_{\mathrm{z}}]_{\mathrm{UAV}} \right\|_2}$$

$$\Omega_{\mathrm{UAV}} = B_{body} * \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

The pseudospectral method seeks a solution that brings the angle of attack from the HUAV to the target, $\Omega_{AngleofAttack} \to \vec{0}$, so that the HUAV is pointing directly at the target at t_f. Additionally, the optimal trajectory is one in which the forward velocity is the same as the initial velocity, x=6 m/s as running fire is more accurate than stationary fire as explained in the United States Army Helicopter Gunnery Manual [18].

III. SOLUTION APPROACH

A. BUILDING THE LIBRARY

1. Finding Optimal Trajectories

Optimal trajectories which minimized the time to reorient the HUAV from its initial state to pointing directly at the target were computed over a grid of targets in ten meter increments from zero to 50 in both the North and East directions. This large range of possible solutions is the most challenging in terms of optimal control. For all targets, the Down coordinate was -2 meters in order to simulate the height of a person or vehicle on level ground. The grid of targets is shown in the NED frame in Figure 4.

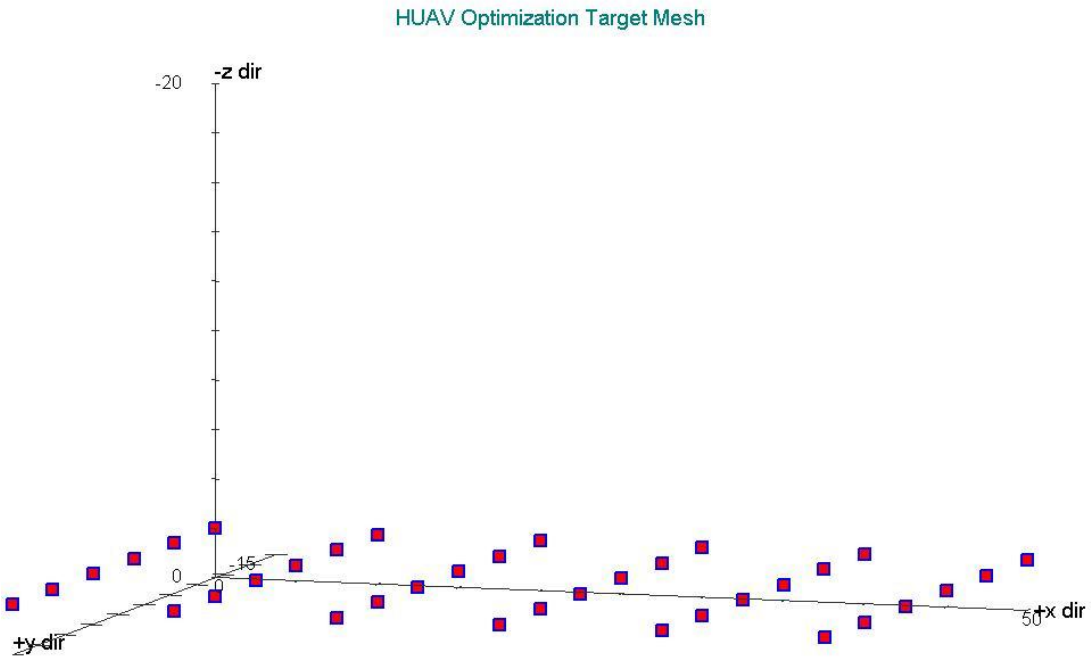


Figure 4. Target Grid, Optimal Trajectories

2. Pseudospectral Method

For the PS method to calculate the optimal trajectory, an initial guess of the final state and control values is required, from which the algorithm makes use of Legendre-Gauss-Lobatto (LGL) quadrature nodes [14]. The trajectory solution, a function *f(t)* is approximated by Nth order Lagrange polynomials using the an approximation of the derivative of the function at these LGL nodes. Typically, the more nodes required to achieve optimal trajectory solutions implies the difficulty or non-linearity of the trajectory solution. The first guess must be entered manually, but once a trajectory is found, whether optimal or not, the calculated trajectory can then be used as the initial guess for subsequent refinements. It should be noted that the PS method can take anywhere from a few seconds to 45 minutes to calculate a solution, based on the quality of the initial guess and resulting smoothness of the optimal trajectory. For all scenarios of the library, an initial guess was provided from a neighboring optimal trajectory, and the number of LGL nodes was increased from ten to forty until an optimal trajectory was achieved.

An example of an optimal trajectory, one that re-orients the HUAV from the initial state to the final heading pointed directly at the target, in minimum time, is shown in Figure 5 as it approaches its final position. The blue vector shows the optimal heading (straight line from the HUAV's current position to the target), while the red vector shows the HUAV's current heading along the ADL. The dotted gray line is the path that the HUAV has flown from initial state to current.

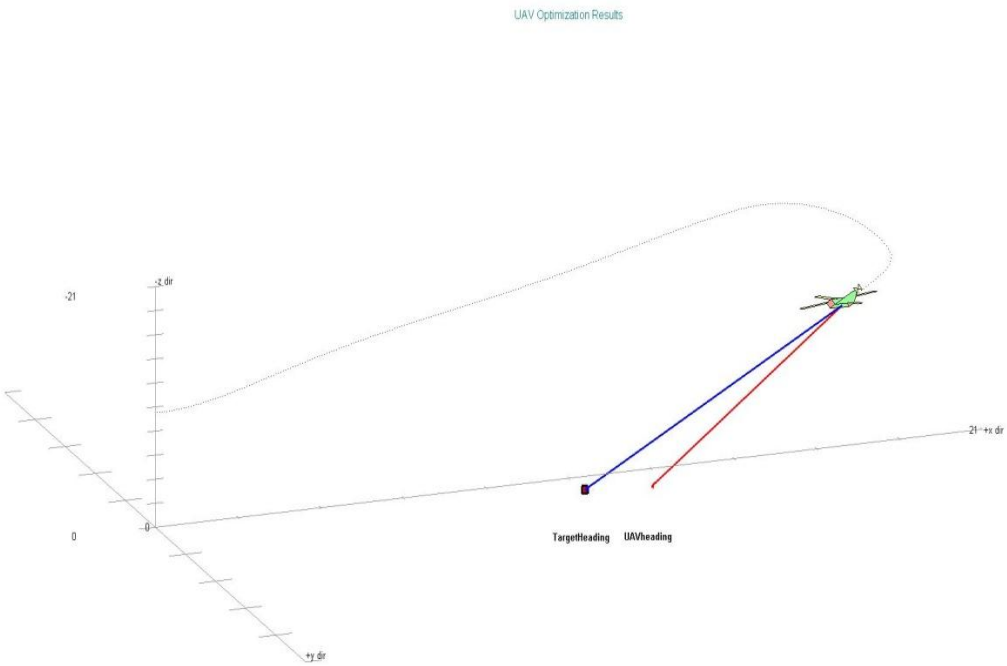


Figure 5. Optimal trajectory, target at [20,5,-2] NED frame

B. INTERPOLATION

1. One Dimension

Once the library of optimal trajectories is populated in the ten meter grid, linear interpolation between known optimal trajectories is proposed in order to approximate an optimal trajectory in less time. To verify the effectiveness of this idea, control inputs from the known optimal trajectories in the library are interpolated to provide control inputs to re-orient to targets in one meter intervals between the known trajectories in the library. The HeLion model was then used to evaluate how close the interpolated control inputs brought the final state of the HUAV to an optimal trajectory.

Given a test target (tgt) between two known targets from the library, *a* and *b*, a linear combination of the two

known optimal trajectories is cubically splined together in the time spectrum based on the linear ratio of the distances between the target being interpolated and known optimal trajectories. The equation below shows how the linear coefficients α and β were calculated based on the distance in one axis, here the North axis.

$$\alpha = \frac{b_x - tgt_x}{\text{interval}} \text{ and } \beta = \frac{tgt_x - a_x}{\text{interval}}$$

The linear coefficients of the known trajectories were then used to provide a guess of the time required for the interpolated trajectory to reorient the HUAV to the final heading, as a ratio of the final time needed for the two known optimal trajectories.

$$t_{tgt,\,final} = \alpha * t_{a,\,final} + \beta * t_{b,\,final}$$

Before interpolation can occur, time step scaling factors need to be calculated to ensure the interpolation of each known trajectory occurs in the correct relative portion of the interpolated trajectory. The following scaling factors were calculated so that at the initial and final times of the interpolated trajectory matched with the initial and final times of the known optimal trajectories being interpolated, and adequately spaced in between, because the final time for each known optimal trajectory was different.

$$k_a = \frac{t_{a,\,final}}{t_{tgt,\,final}} \quad \text{and} \quad k_b = \frac{t_{b,\,final}}{t_{tgt,\,final}}$$

Finally, the approximated control inputs were computed using a linear ratio of the two known trajectories control inputs, and MATLAB's ODE45 solver was used to provide a

numerical integration solution. The solver integrated the differential equation relating the change in the state vector as a function of the control inputs and current state vector, $\dot{x} = f(x, u)$, as provided through the HeLion model.

$$u_{tgt} = \alpha(\text{interpolated } u_a \text{ at } k_a t) + \beta(\text{interpolated } u_b \text{ at } k_b t)$$

Once the control inputs were interpolated and integrated to calculate the states of the HUAV, the trajectory was evaluated by computing the angular error, $\Omega_{AngleofAttack,error}$.

$$\Omega_{AngleofAttack,error} = \cos^{-1}\left(\Omega_{UAV,final}{}^{T} * \Omega_{TGT,final}\right) * \frac{180}{\pi}$$

2. Two Dimensions

Bilinear interpolation was used to approximate solutions based on the nearest four known optimal trajectories. Given a target location, the two-dimension interpolation routine located the two nearest optimal trajectories in the North coordinate axis, and the two nearest neighbors in the East coordinate axis as seen in Figure 6. Using the ten meter library it is possible to interpolate a target trajectory from $North, East \in \{o\ to\ 50\}$, where each possible target location is less than ten meters in both the North and East directions from a known optimal trajectory.

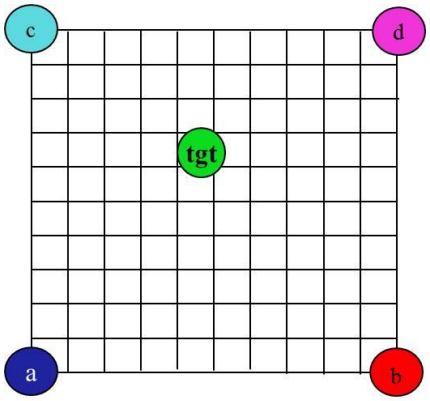


Figure 6. Two Dimension Bi-Linear Interpolation Grid

Then, it takes the product of the two linear ratios between the two nearest neighbors, divided by the interval width. The ratio for each neighboring point used in interpolation is in the range of zero to one. The ratio value is exactly one if the point happens to fall on a known optimal trajectory, and if the point is exactly in the middle of the four nearest optimal trajectories, each of the four ratios is exactly 25%.

$$\alpha = \frac{(b_x - tgt_x)*(c_y - tgt_y)}{\text{interval}^2},\ \beta = \frac{(tgt_x - a_x)*(d_y - tgt_y)}{\text{interval}^2}$$

$$\gamma = \frac{(d_x - tgt_x)*(tgt_y - a_y)}{\text{interval}^2},\ \delta = \frac{(tgt_x - c_x)*(tgt_y - b_y)}{\text{interval}^2}$$

The ratios were used to estimate the final time for the interpolated trajectory, and scaling factors used for the interpolation table lookup were used in the same manner as the one dimension case, given by

$$t_{tgt,final} = \alpha \cdot t_{a,final} + \beta \cdot t_{b,final} + \gamma \cdot t_{c,final} + \delta \cdot t_{d,final}$$

$$k_i = \frac{t_{i,final}}{t_{tgt,final}}, i \in \{a,b,c,d\}$$

The final control vector was again approximated using the cubic spline interpolated values of the four known optimal trajectories, and numerical integration provided the final trajectory solution.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS

A. LIBRARY OF OPTIMAL TRAJECTORIES

The database of known optimal trajectories distributed over a 50 by 50 meter grid in ten meter intervals is shown in Table 4. At each grid location, the exit code is shown first, which gives a measure of how well the provided solution meets the performance cost and constraints. A value of one is best, meaning a solution was found satisfying optimality conditions, while a code of 41 means the program terminated after numerical difficulties and the solution cannot be improved further [19]. There are many other exit codes, but these two are two of the best the program can achieve, so all optimal trajectories utilized for the database were required to achieve a one or 41 exit code.

Along with the exit code, the next value is the number of LGL quadrature nodes required to achieve the best exit code is listed. Typically, the more nodes required to achieve optimal trajectory solutions implies the difficulty or non-linearity of the trajectory solution. The dark green cells indicate an optimal solution with a minimum number of nodes. Lighter green requires more nodes but still achieves an optimal solution, while yellow cells reached a resource limit or could not be improved further, but still satisfied all constraints.

Table 4. Library of optimal trajectories

Target North Coord.	0	10	20	30	40	50
50	1, 20 nodes	1, 20 nodes	1, 20 nodes	1, 30 nodes	1, 30 nodes	1, 10 nodes
40	1, 20 nodes	1, 10 nodes	1, 20 nodes	1, 30 nodes	1, 10 nodes	1, 10 nodes
30	1, 10 nodes	41, 10 nodes	1, 10 nodes	1, 10 nodes	1, 10 nodes	1, 10 nodes
20	1, 10 nodes	41, 10 nodes	1, 40 nodes	1, 10 nodes	1, 10 nodes	1, 10 nodes
10	1, 10 nodes	1, 20 nodes	1, 40 nodes	1, 10 nodes	1, 20 nodes	1, 10 nodes
0	1, 20 nodes	1, 10 nodes	1, 30 nodes	1, 10 nodes	1, 10 nodes	1, 30 nodes

Target East Coordinate

To illustrate how long it takes to populate the library using the pseudospectral method, the average CPU time is 338.9723 seconds using only ten nodes, with an average final angular error of .4662 degrees.

B. INTERPOLATION

1. One Dimension Interpolation

With the library of optimal trajectories available, we ran the one dimension MATLAB routine to interpolate control inputs between known optimal trajectories spaced at ten meter intervals. To determine the magnitude of error, one meter increments were approximated and the HUAV's final

heading to target as simulated by the HeLion model was compared to the optimal straight line vector from the HUAV's final position to the target.

The final heading to target angular error is easily visualized in Figure 7, where the two optimal trajectories computed via the pseudospectral method are shown in red and blue stars in their discreet form, while they are connected with solid lines representing the continuous solution representation. The interpolated trajectory is shown in green dashed line. The difference between the solid blue vector, representing the straight line, minimum distance vector from the HUAV's position to the target, and the solid red vector, the HUAV's actual heading, depicts the angular error.

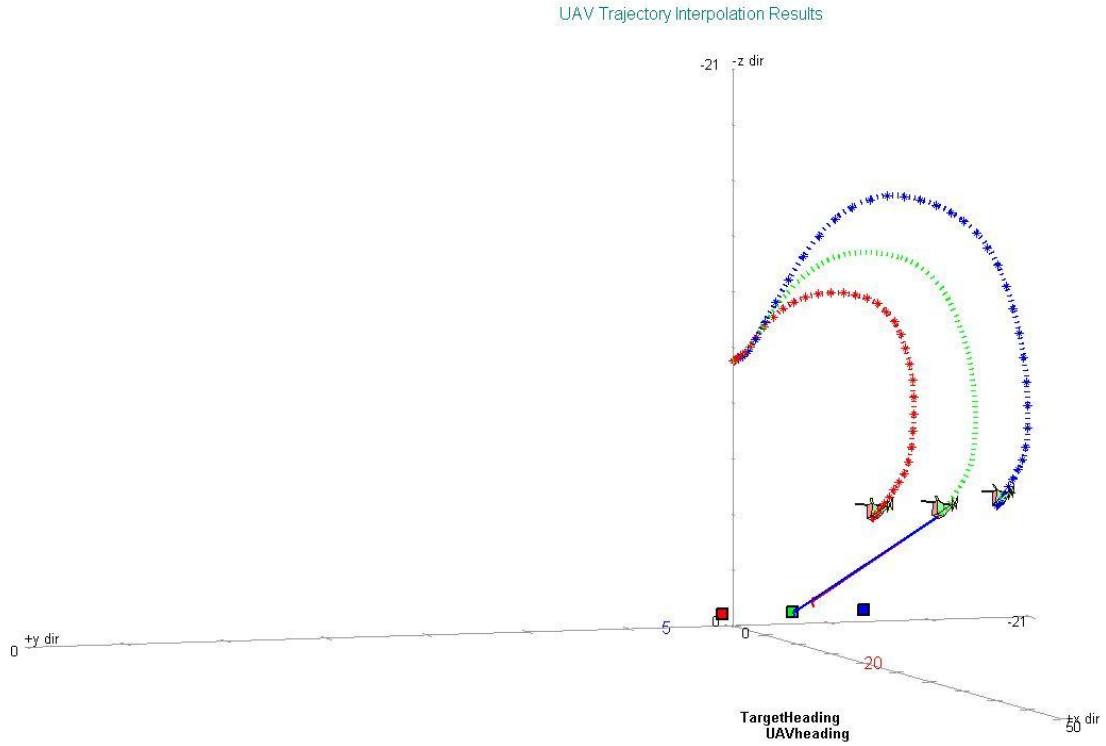


Figure 7. 1D Interpolated trajectory, [20 5-2]East axis view

In this view the error appears minimal. However, when the view is rotated to see the heading in the North direction, the angular error is more apparent, shown in Figure 8. Here, looking down along the North axis, we see the interpolation error which contributes most significantly to the angular error of 6.8449 degrees

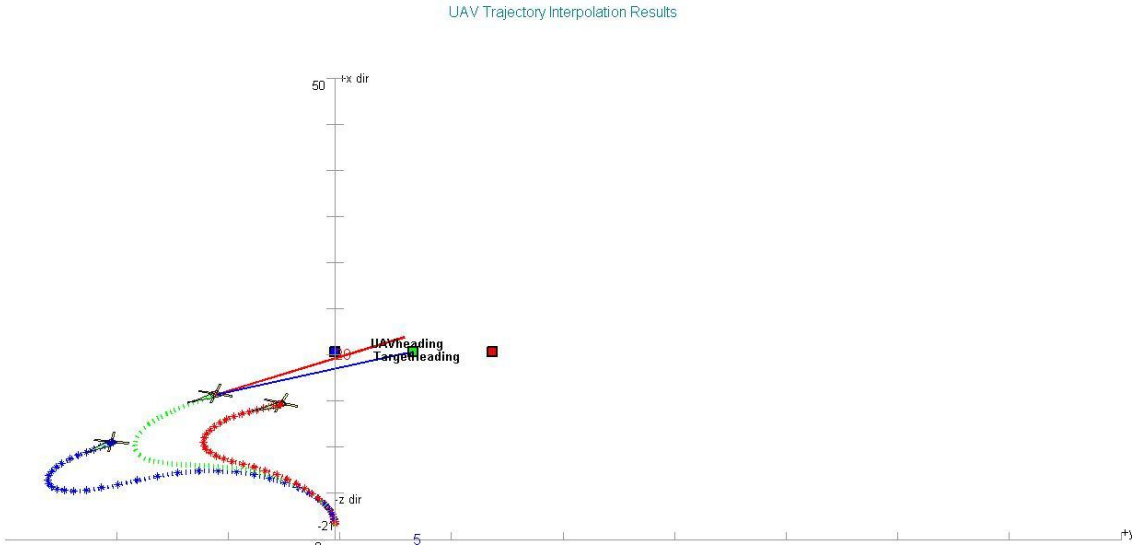


Figure 8. One dimension interpolated trajectory for a target located at [20 5-2]North axis view

As expected, the largest error occurs at the maximum absolute distance between optimal trajectories, the midpoint. For most of the intervals the final angular error is less than the five degree threshold. In fact, interpolations in the Y direction yield an average angular error of 1.6426 degrees, with each interpolation computed in an average of 11.106 seconds as computed by MATLAB.

However, there are disturbing intervals where the error exceeds the threshold, notably when the target is 20 meters in front of the HUAV and between ten to 20 meters to

the right. Interpolations in this region yield angular error values in excess of 18 degrees, shown highlighted in red in Figure 9.

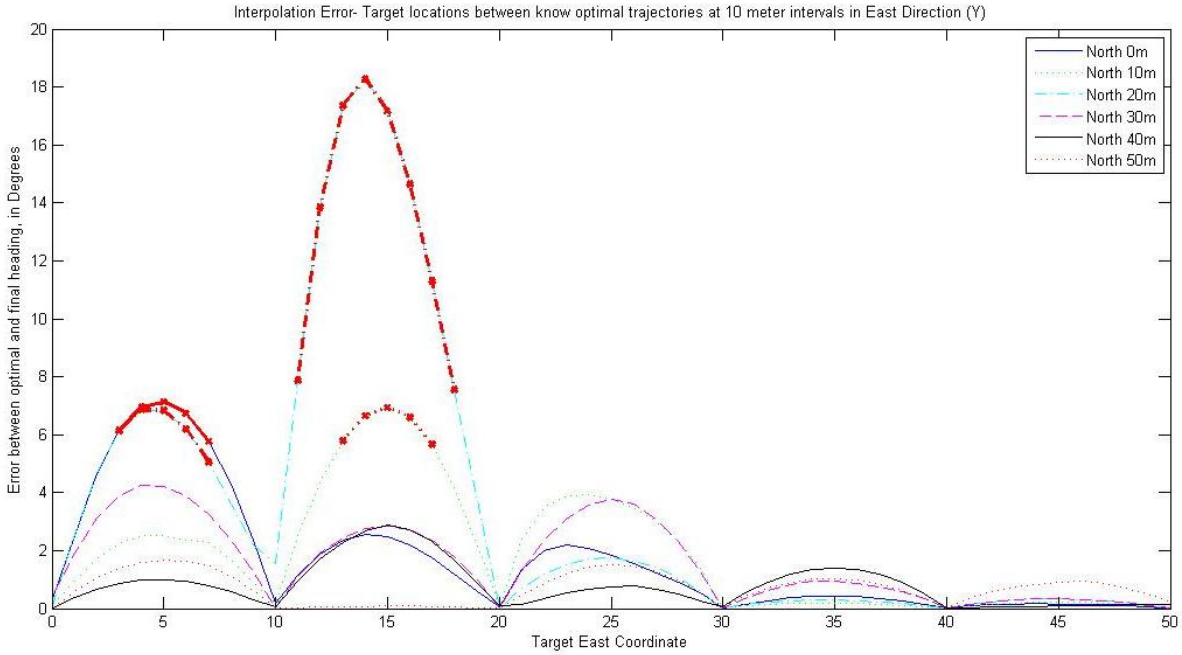


Figure 9. Final HUAV heading to target angular error, one dimension interpolation in East axis

The interpolation error is worse when the target's East coordinate is held constant and trajectories are interpolated in the North direction. This makes sense as the initial guesses for the trajectory library were provided by the nearest neighbor in the East direction. Therefore, trajectories for targets with the same East coordinate are more similar than neighbors in the North direction. So when trajectories are interpolated in one meter intervals in the North axis, there is a less similarity and therefore larger angular error as seen in Figure 10. Average angular error for this scenario is 3.2734 degrees, with an average time to compute of 11.6253 seconds.

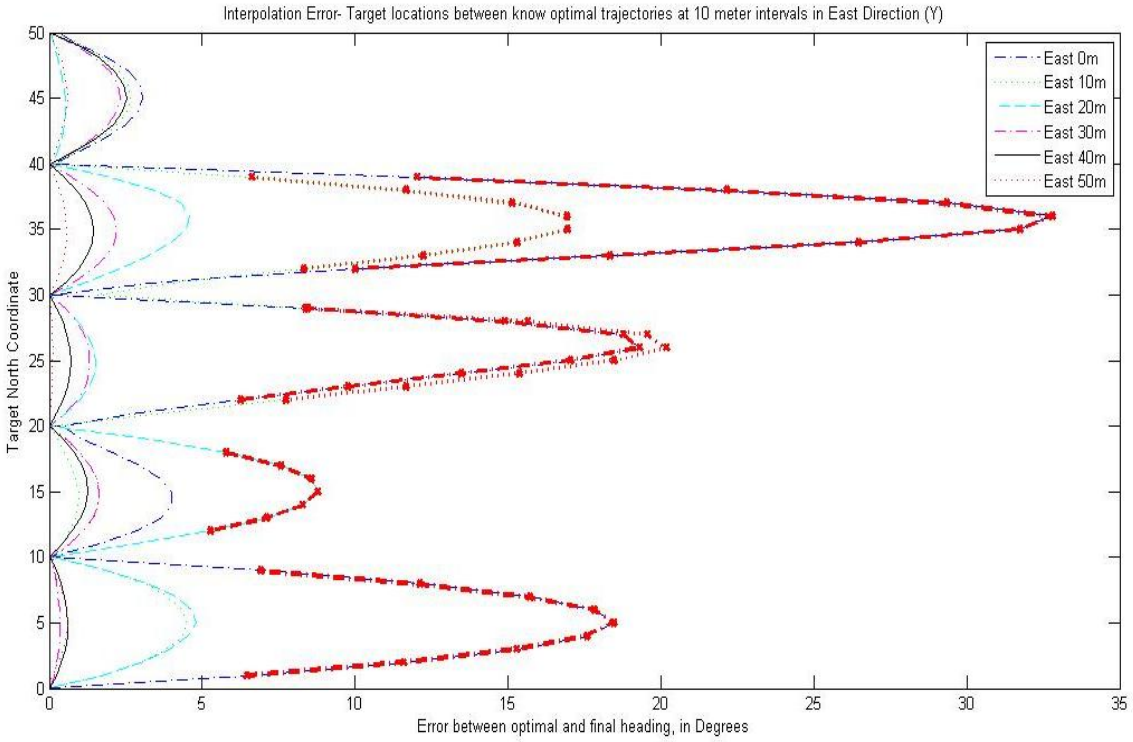


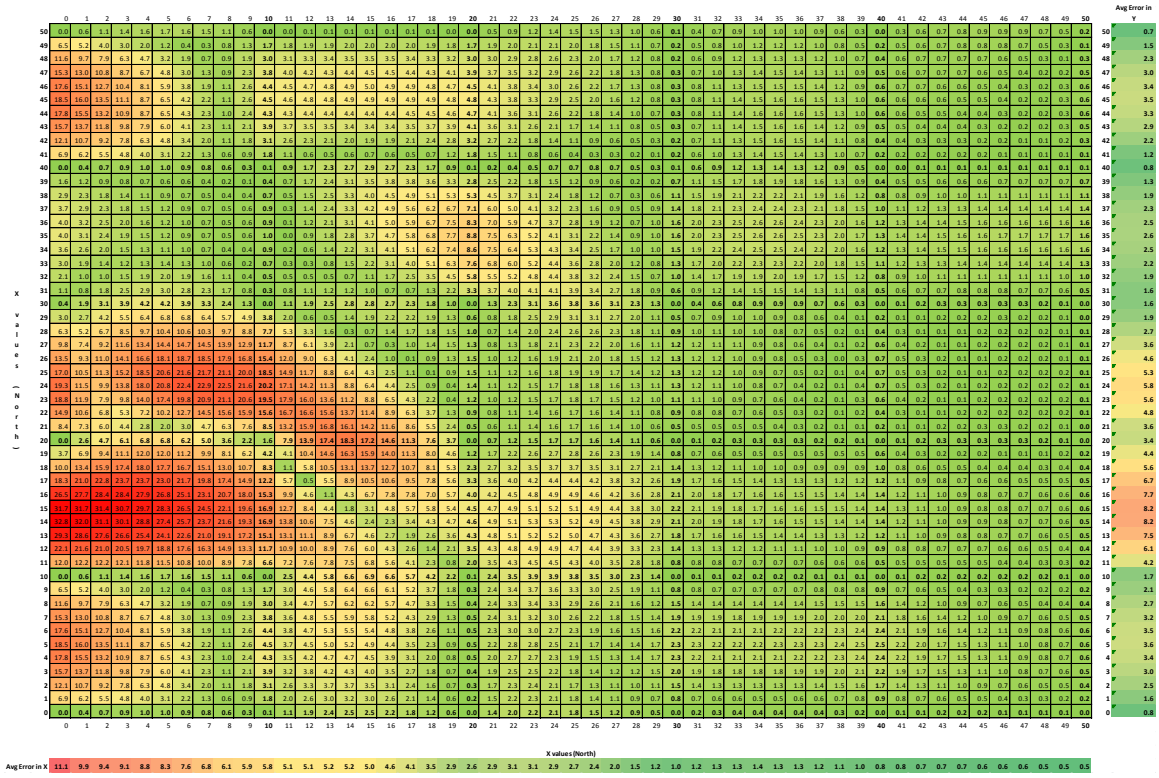
Figure 10. Final HUAV heading to target angular error, one dimension interpolation in North axis

2. Two Dimension Interpolation

Results for the two dimension interpolation were as expected for target locations greater than 30 meters away from the HUAV initial position in both the North and East coordinates. However, there were some interesting results at ranges between 10 to 30 meters in the North direction and less than 20 meters in the East direction, as well as North from 40 to 50 with East less than 5 meters. In these two regions, the angular error between the final HUAV heading and optimal heading greatly exceeded the five degree threshold. The green shaded areas in Table 5 depict angular errors less than the threshold. Cells shaded yellow

indicate marginal error values between five and ten degrees, while the red highlight where the error is unacceptable, above 20 degrees.

Table 5. Two-dimension interpolation final heading to target angular error, using the 10 meter library



The angular error between the optimal and actual heading at the interpolated trajectory's final state is also depicted in the surface plot in Figure 11.

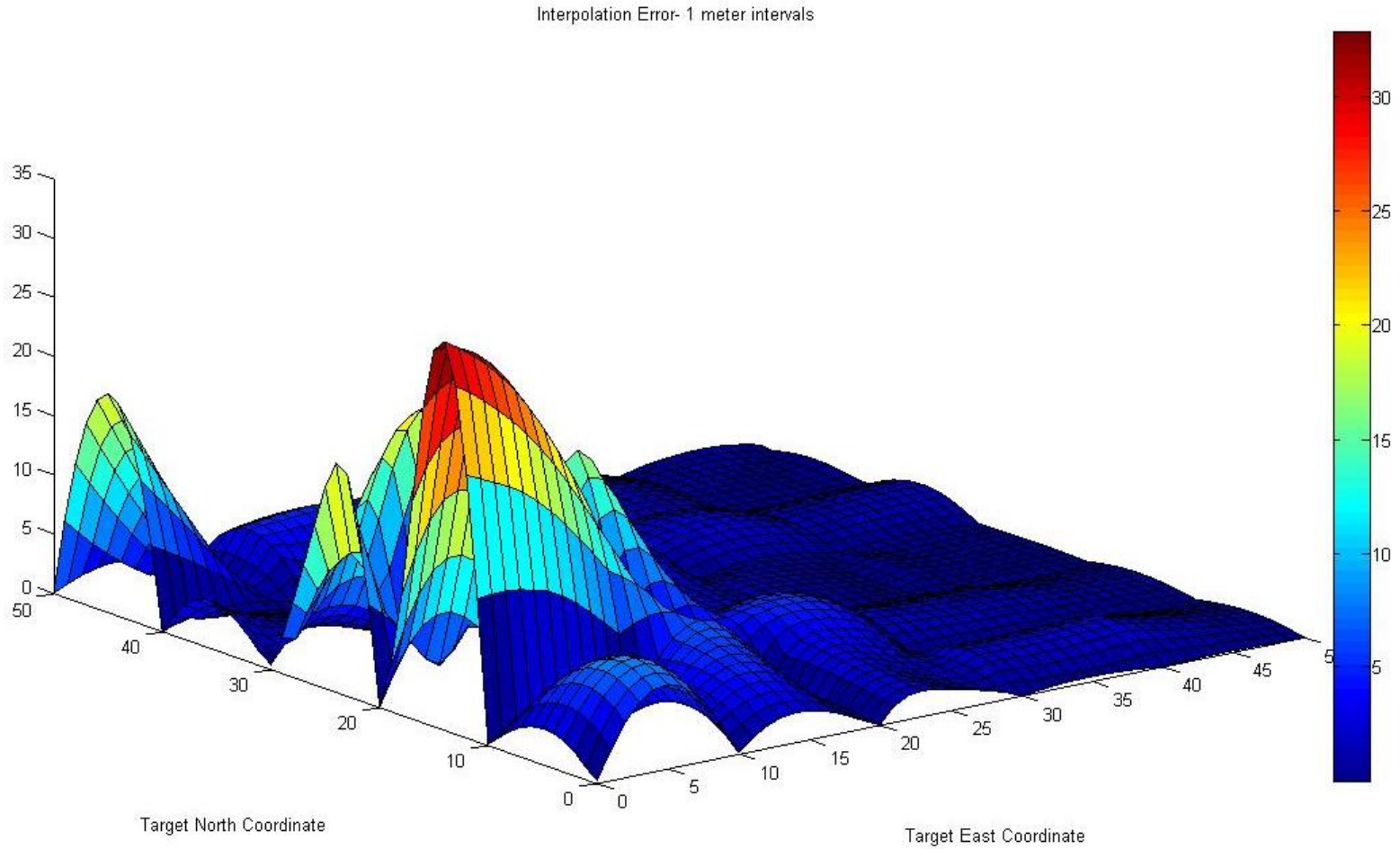


Figure 11. Final angular error between the optimal and actual heading at the interpolated trajectory's final state

These results made it evident that there was unacceptable error in the regions with red shading. These regions occur in areas where there seems to be a bifurcation point in the optimal trajectory paths. The bifurcation areas in this sense are when the HUAV optimal trajectory shifts suddenly from one family of trajectory types to another. For instance, when the target is within close range to the HUAV's initial position, the optimal trajectory requires the HUAV to fly *away* from the target initially to allow the descent required to point the HUAV at the target without exceeding aircraft limitations. At t_f the aircraft is actually pointing in almost the opposite direction of its initial state. This flight characteristic continues until the target is roughly 10 meters from the origin, at which point the optimal trajectories quickly change to allow the HUAV to fly a more teardrop shape, where the helicopter will turn away slightly in order to most rapidly descend to a sufficient altitude to put the sensor within limits. The region where the trajectories switch from these different path families is where large error develops in the interpolation method.

The largest error occurs between 20 to 30 meters to the target, when the trajectories again shift to one that does not require it to fly away or descend much, but rather turn and point directly at the target. The final angular error is largest in this region because of the difference in the minimal time required to re-orient. At close range, the HUAV spends more time descending and even turning away from the target, which leads to a larger t_f.

To reduce the error, the interpolation interval was reduced from ten meters to five meters in the regions with large error. Then, the angular error was recomputed in 1 meter increments in these regions. The reduced final angular error is clearly evident comparing Figures 12 through 15 to Figure 11.

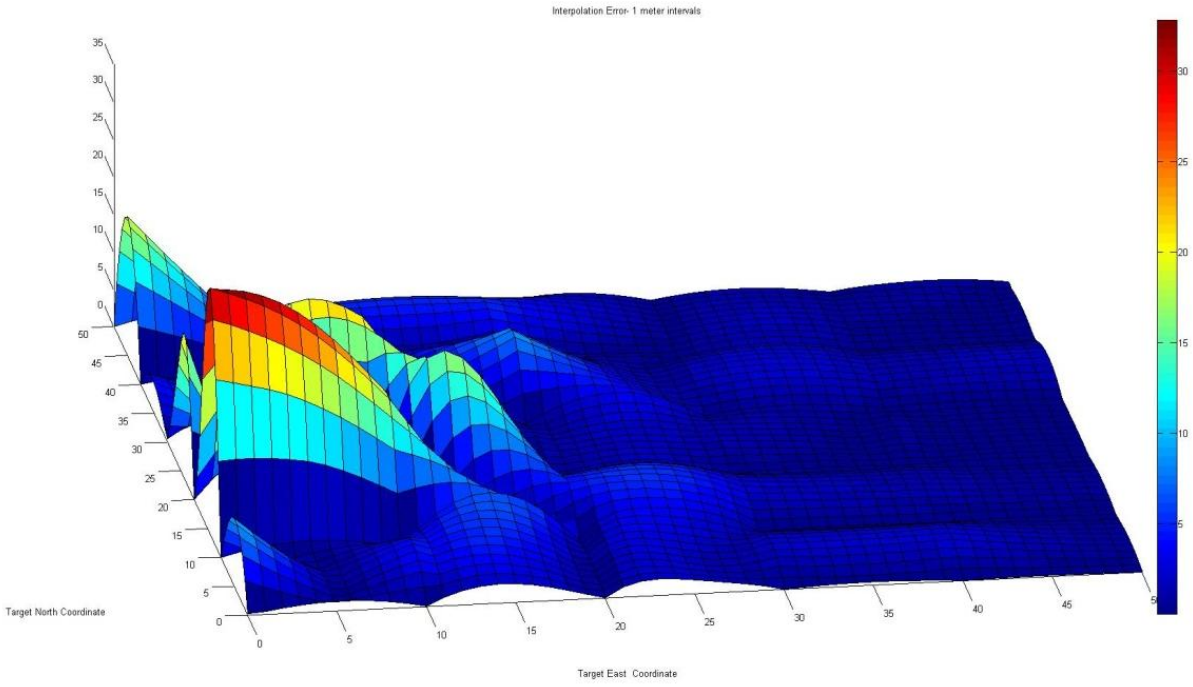


Figure 12. Final angular error with additional 5 meter grid from North∈{0 to 10 }and East∈{0 to 10}

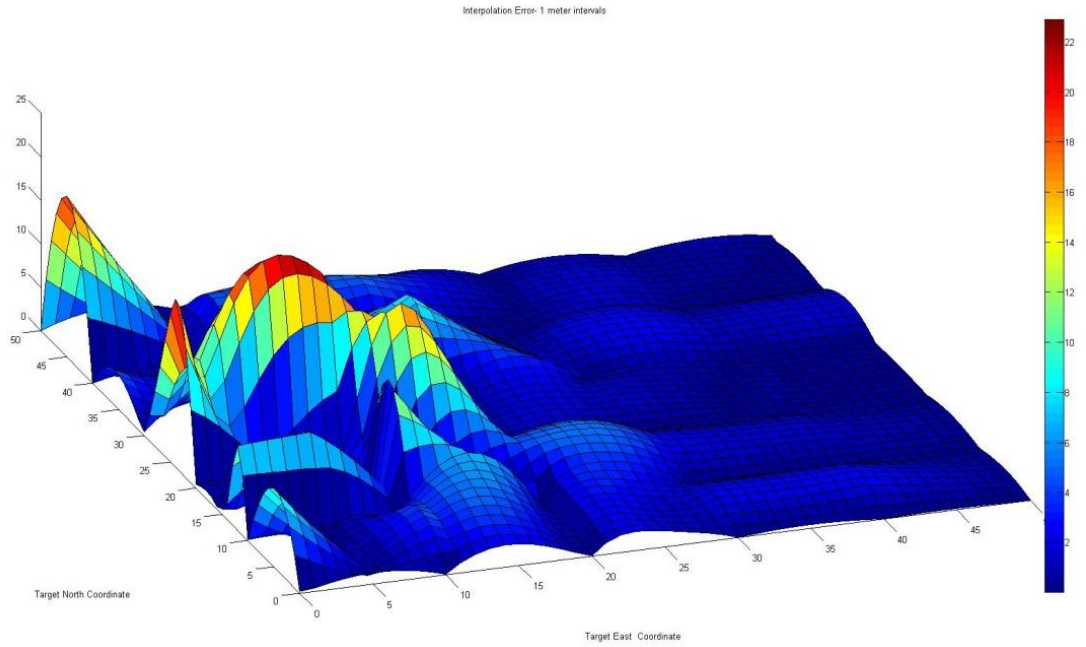


Figure 13. Final angular error with additional 5 meter grid from North∈{10 to 20 }and East∈{0 to 10}

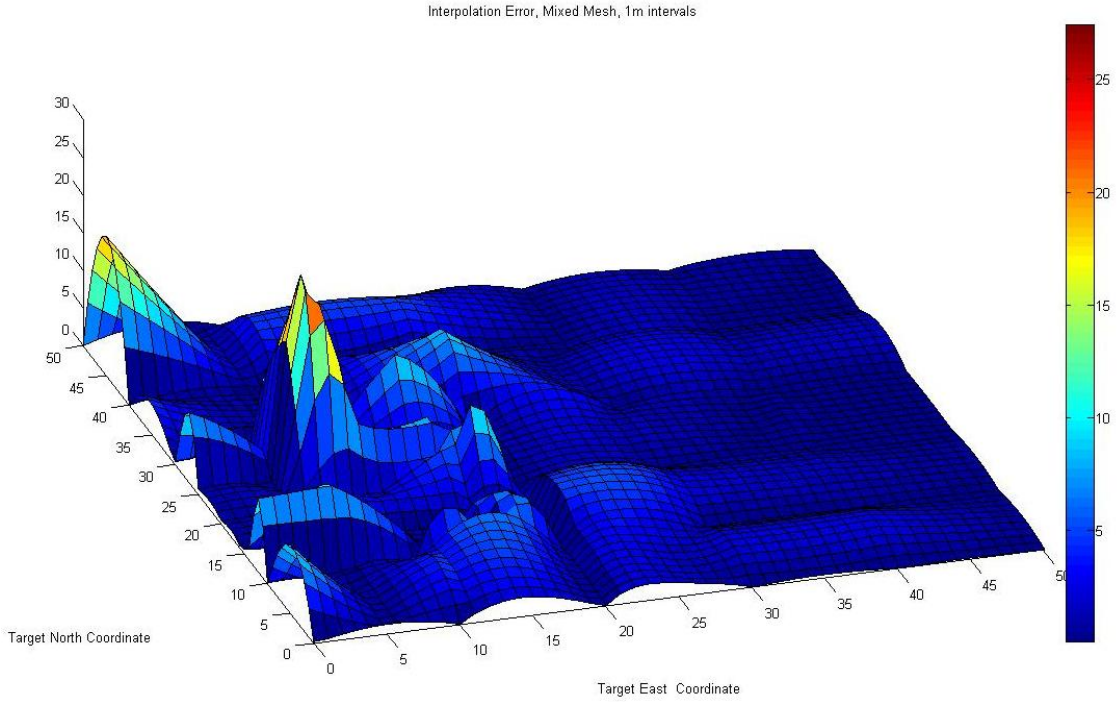


Figure 14. Final angular error with additional 5 meter grid from North∈{10 to 30 }and East∈{0 to 20}

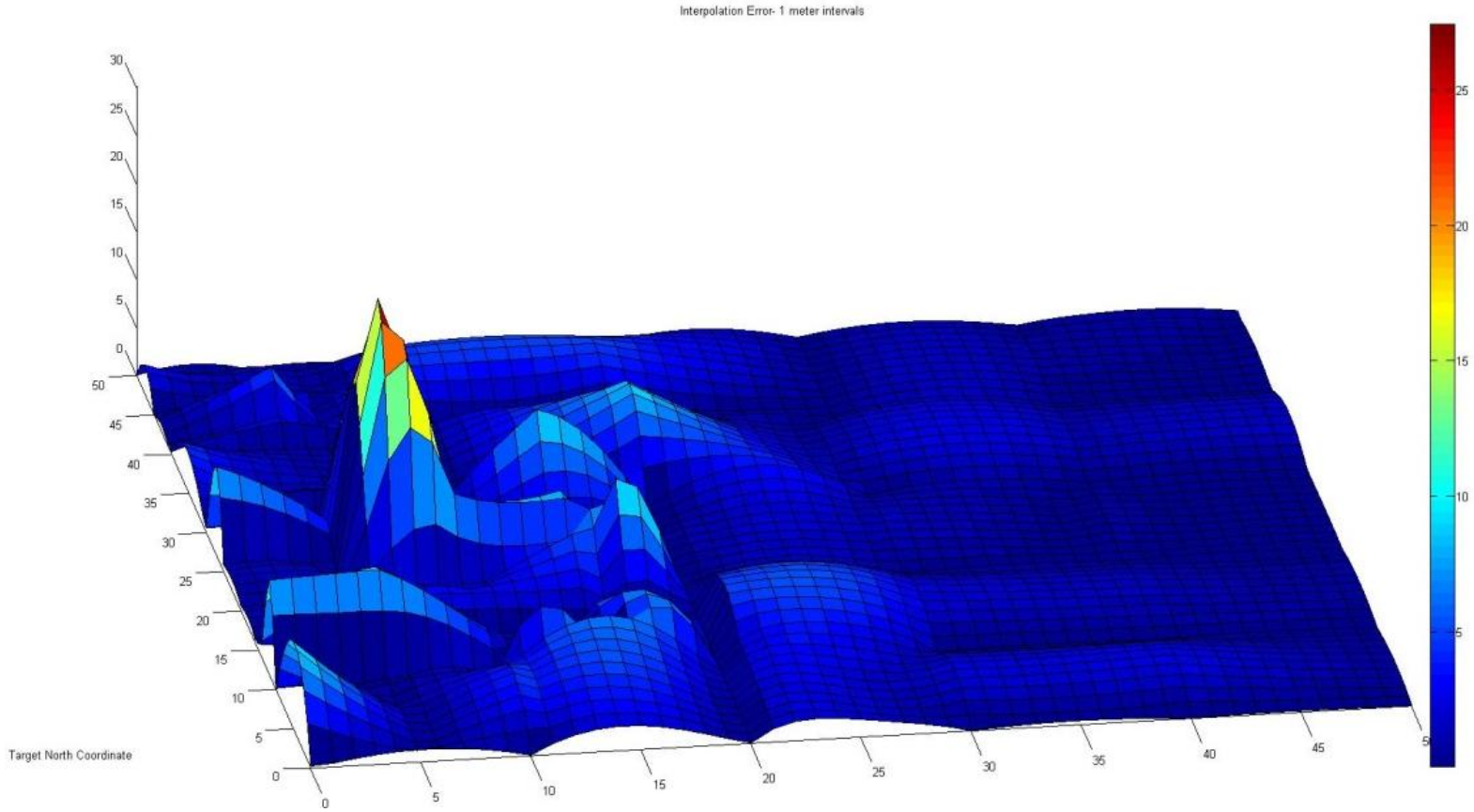


Figure 15. Final angular error with additional 5 meter grid from North∈{0 to 50 }and East∈{0 to 20}

Table 6. Final angular errors using the mixed mesh Library

X values (North)

Avg Error in Y	Y values (East)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
	Avg Error in X	3.2	2.8	2.6	2.3	2.1	1.8	3.9	5.0	4.7	3.5	2.2	2.3	2.8	3.2	3.5	3.9	4.0	4.1	3.7	2.8	1.9	2.9	3.1	3.1	2.9	2.7	2.4	2.0	1.5	1.2	1.0	1.2	1.3	1.3	1.4	1.3	1.3	1.2	1.1	1.0	0.8	0.8	0.7	0.7	0.7	0.6	0.6	0.6	0.5	0.5	0.5
0.5	50	0.0	0.5	0.8	0.8	0.5	0.0	0.2	0.3	0.3	0.2	0.0	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.5	0.9	1.2	1.4	1.5	1.5	1.3	1.0	0.6	0.1	0.4	0.7	0.9	1.0	1.0	1.0	0.9	0.6	0.3	0.0	0.3	0.6	0.7	0.8	0.9	0.9	0.9	0.7	0.5	0.2
1.2	49	1.8	1.2	0.8	0.7	0.8	1.1	0.6	0.3	0.6	1.2	1.8	1.8	1.9	1.9	2.0	2.0	2.0	2.0	1.9	1.8	1.7	1.9	2.0	2.1	2.1	2.0	1.8	1.5	1.1	0.7	0.2	0.5	0.8	1.0	1.2	1.2	1.2	1.0	0.8	0.5	0.2	0.5	0.6	0.7	0.8	0.8	0.8	0.7	0.5	0.3	0.1
1.7	48	2.7	2.0	1.5	1.2	1.2	1.4	0.9	0.5	0.8	1.7	2.7	3.1	3.3	3.4	3.5	3.5	3.5	3.4	3.3	3.2	3.0	3.0	2.9	2.8	2.6	2.3	2.0	1.7	1.2	0.8	0.2	0.6	0.9	1.2	1.3	1.3	1.3	1.2	1.0	0.7	0.4	0.6	0.7	0.7	0.7	0.7	0.6	0.5	0.3	0.1	0.3
1.9	47	2.8	2.0	1.5	1.2	1.2	1.4	1.0	0.4	0.4	1.5	2.7	4.0	4.2	4.3	4.4	4.5	4.5	4.4	4.3	4.1	3.9	3.7	3.5	3.2	2.9	2.6	2.2	1.8	1.3	0.8	0.3	0.7	1.0	1.3	1.4	1.5	1.4	1.3	1.1	0.9	0.5	0.6	0.7	0.7	0.7	0.6	0.5	0.4	0.2	0.2	0.5
1.9	46	1.9	1.2	0.8	0.6	0.6	0.8	1.0	0.8	0.3	0.6	1.8	4.5	4.7	4.8	4.9	5.0	4.9	4.9	4.8	4.7	4.5	4.1	3.8	3.4	3.0	2.6	2.2	1.7	1.3	0.8	0.3	0.8	1.1	1.3	1.5	1.5	1.5	1.4	1.2	0.9	0.6	0.7	0.7	0.6	0.6	0.5	0.4	0.3	0.2	0.3	0.6
1.9	45	0.0	0.5	0.7	0.7	0.6	0.3	0.8	1.3	1.3	0.9	0.0	4.6	4.8	4.8	4.9	4.9	4.9	4.9	4.9	4.8	4.8	4.3	3.8	3.3	2.9	2.5	2.0	1.6	1.2	0.8	0.3	0.8	1.1	1.4	1.5	1.6	1.6	1.5	1.3	1.0	0.6	0.6	0.6	0.6	0.5	0.5	0.4	0.2	0.2	0.3	0.6
2.0	44	0.4	1.2	2.1	2.9	3.6	4.3	1.9	0.3	1.3	2.1	2.4	4.3	4.4	4.4	4.4	4.4	4.4	4.5	4.5	4.6	4.7	4.1	3.6	3.1	2.6	2.2	1.8	1.4	1.0	0.7	0.3	0.8	1.1	1.4	1.6	1.6	1.6	1.5	1.3	1.0	0.6	0.6	0.5	0.5	0.4	0.4	0.3	0.2	0.2	0.3	0.6
2.0	43	0.7	1.4	2.5	3.7	4.8	5.9	3.0	0.8	1.5	2.9	3.8	3.7	3.5	3.5	3.4	3.4	3.4	3.5	3.7	3.9	4.1	3.6	3.1	2.6	2.1	1.7	1.4	1.1	0.8	0.5	0.3	0.7	1.1	1.4	1.5	1.6	1.6	1.4	1.2	0.9	0.5	0.5	0.4	0.4	0.4	0.3	0.2	0.2	0.2	0.3	0.5
1.5	42	0.6	1.1	2.1	3.1	4.2	5.4	2.7	0.6	1.5	2.8	3.8	2.6	2.3	2.1	2.0	1.9	1.9	2.1	2.4	2.8	3.2	2.7	2.2	1.8	1.4	1.1	0.9	0.6	0.5	0.3	0.2	0.7	1.1	1.3	1.5	1.6	1.5	1.4	1.1	0.8	0.4	0.4	0.3	0.3	0.3	0.2	0.2	0.1	0.1	0.2	0.3
0.8	41	0.3	0.5	0.9	1.4	1.9	2.5	0.9	0.5	1.4	2.1	2.5	1.1	0.6	0.5	0.6	0.7	0.6	0.5	0.7	1.2	1.8	1.5	1.1	0.8	0.6	0.4	0.3	0.3	0.2	0.1	0.2	0.6	1.0	1.3	1.4	1.5	1.4	1.3	1.0	0.7	0.2	0.2	0.2	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.1
1.0	40	0.2	0.9	1.5	1.9	2.3	2.6	2.3	1.9	1.3	0.7	0.1	0.9	1.7	2.3	2.7	2.9	2.7	2.3	1.7	0.9	0.1	0.2	0.4	0.5	0.7	0.7	0.8	0.7	0.5	0.3	0.1	0.6	0.9	1.2	1.3	1.4	1.3	1.2	0.9	0.5	0.0	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
1.3	39	1.6	1.2	0.9	0.8	0.7	0.6	0.6	0.4	0.2	0.1	0.4	0.7	1.7	2.4	3.1	3.5	3.8	3.8	3.6	3.3	2.8	2.5	2.2	1.8	1.5	1.2	0.9	0.6	0.2	0.2	0.7	1.1	1.5	1.7	1.8	1.9	1.8	1.6	1.3	0.9	0.4	0.5	0.5	0.6	0.6	0.6	0.7	0.7	0.7	0.7	0.7
1.9	38	2.9	2.3	1.8	1.4	1.1	0.9	0.7	0.5	0.4	0.4	0.7	0.5	1.5	2.5	3.3	4.0	4.5	4.9	5.1	5.3	5.3	4.5	3.7	3.1	2.4	1.8	1.2	0.7	0.3	0.6	1.1	1.5	1.9	2.1	2.2	2.2	2.1	1.9	1.6	1.2	0.8	0.8	0.9	1.0	1.0	1.1	1.1	1.1	1.1	1.1	1.1
2.3	37	3.7	2.9	2.3	1.8	1.5	1.2	0.9	0.7	0.5	0.6	0.9	0.3	1.4	2.4	3.3	4.2	4.9	5.6	6.2	6.7	7.1	6.0	5.0	4.1	3.2	2.3	1.6	0.9	0.5	0.9	1.4	1.8	2.1	2.3	2.4	2.4	2.3	2.1	1.8	1.5	1.0	1.1	1.2	1.3	1.3	1.4	1.4	1.4	1.4	1.4	1.4
2.5	36	4.0	3.2	2.5	2.0	1.6	1.2	1.0	0.7	0.5	0.6	0.9	0.1	1.2	2.1	3.1	4.1	5.0	5.9	6.7	7.5	8.3	7.0	5.9	4.7	3.7	2.8	1.9	1.2	0.7	1.0	1.6	2.0	2.3	2.5	2.6	2.6	2.4	2.3	2.0	1.6	1.2	1.3	1.4	1.4	1.5	1.6	1.6	1.6	1.6	1.6	1.6
2.6	35	4.0	3.1	2.4	1.9	1.5	1.2	0.9	0.7	0.5	0.6	1.0	0.0	0.9	1.8	2.8	3.7	4.7	5.8	6.8	7.7	8.8	7.5	6.3	5.2	4.1	3.1	2.2	1.4	0.9	1.0	1.6	2.0	2.3	2.5	2.6	2.6	2.5	2.3	2.0	1.7	1.3	1.4	1.4	1.5	1.6	1.6	1.7	1.7	1.7	1.7	1.6
2.5	34	3.6	2.6	2.0	1.5	1.3	1.1	1.0	0.7	0.4	0.4	0.9	0.2	0.6	1.4	2.2	3.1	4.1	5.1	6.2	7.4	8.6	7.5	6.4	5.3	4.3	3.4	2.5	1.7	1.0	1.0	1.5	1.9	2.2	2.4	2.5	2.5	2.4	2.2	2.0	1.6	1.2	1.3	1.4	1.5	1.5	1.6	1.6	1.6	1.6	1.6	1.6
2.2	33	3.0	1.9	1.4	1.2	1.3	1.4	1.3	1.0	0.6	0.2	0.7	0.3	0.3	0.8	1.5	2.2	3.1	4.0	5.1	6.3	7.6	6.8	6.0	5.2	4.4	3.6	2.8	2.0	1.2	0.8	1.3	1.7	2.0	2.2	2.3	2.3	2.2	2.0	1.8	1.5	1.1	1.2	1.3	1.3	1.4	1.4	1.4	1.4	1.4	1.4	1.3
1.9	32	2.1	1.0	1.0	1.5	1.9	2.0	1.9	1.6	1.1	0.4	0.5	0.5	0.5	0.5	0.7	1.1	1.7	2.5	3.5	4.5	5.8	5.5	5.2	4.8	4.4	3.8	3.2	2.4	1.5	0.7	1.0	1.4	1.7	1.9	1.9	2.0	1.9	1.7	1.5	1.2	0.8	0.9	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.0	1.0
1.6	31	1.1	0.8	1.8	2.5	2.9	3.0	2.8	2.3	1.7	0.8	0.3	0.8	1.1	1.2	1.2	1.0	0.7	0.7	1.3	2.2	3.3	3.7	4.0	4.1	4.1	3.9	3.4	2.7	1.8	0.9	0.6	0.9	1.2	1.4	1.5	1.5	1.4	1.3	1.1	0.8	0.5	0.6	0.7	0.7	0.8	0.8	0.8	0.7	0.7	0.6	0.5
2.7	30	0.4	0.5	0.9	1.0	0.7	0.4	10.0	12.6	10.8	6.2	0.0	2.4	4.7	6.9	8.9	10.8	10.0	8.4	6.2	3.4	0.0	1.3	2.3	3.1	3.6	3.8	3.6	3.1	2.3	1.3	0.0	0.4	0.6	0.8	0.9	0.9	0.9	0.7	0.6	0.3	0.0	0.1	0.2	0.3	0.3	0.3	0.3	0.3	0.2	0.1	0.0
2.9	29	4.7	4.5	4.1	3.5	2.6	1.4	12.2	15.5	13.1	7.5	0.6	1.5	3.4	5.2	6.8	8.3	7.9	6.9	5.2	2.9	0.1	0.8	1.8	2.5	2.9	3.1	3.1	2.7	2.0	1.1	0.5	0.7	0.9	1.0	1.0	0.9	0.8	0.6	0.4	0.1	0.2	0.1	0.1	0.2	0.2	0.2	0.3	0.2	0.2	0.1	0.0
3.0	28	8.2	7.4	6.4	5.3	4.0	2.6	14.4	18.6	15.3	8.6	1.0	0.9	2.3	3.6	4.8	5.9	6.0	5.4	4.2	2.4	0.1	0.7	1.4	2.0	2.4	2.6	2.6	2.3	1.8	1.1	0.9	1.0	1.1	1.0	1.0	0.8	0.7	0.5	0.2	0.1	0.4	0.3	0.1	0.1	0.1	0.2	0.2	0.2	0.2	0.1	0.1
3.0	27	9.3	8.4	7.3	6.0	4.7	3.1	16.3	21.7	17.6	9.4	1.2	0.8	1.5	2.3	3.0	3.6	4.1	3.9	3.1	1.8	0.1	0.8	1.3	1.8	2.1	2.3	2.2	2.0	1.6	1.1	1.2	1.2	1.1	1.1	0.9	0.8	0.6	0.4	0.1	0.2	0.6	0.4	0.2	0.1	0.1	0.2	0.2	0.2	0.2	0.1	0.1
2.8	26	6.8	6.4	5.7	4.8	3.7	2.3	17.8	24.7	19.7	9.9	1.4	1.0	1.1	1.4	1.5	1.5	2.3	2.5	2.2	1.2	0.3	1.0	1.2	1.6	1.9	2.1	2.0	1.8	1.5	1.2	1.3	1.2	1.2	1.0	0.9	0.8	0.5	0.3	0.0	0.3	0.7	0.5	0.3	0.2	0.1	0.1	0.2	0.2	0.2	0.1	0.1
2.3	25	0.7	0.9	0.9	0.7	0.3	0.4	18.8	27.4	21.4	9.8	1.8	1.6	1.3	0.9	0.4	0.3	0.9	1.4	1.4	0.8	0.5	1.1	1.2	1.6	1.8	1.9	1.9	1.7	1.4	1.2	1.3	1.2	1.2	1.0	0.9	0.7	0.5	0.2	0.1	0.4	0.7	0.5	0.3	0.2	0.1	0.1	0.2	0.2	0.2	0.2	0.1
2.6	24	0.9	1.2	1.3	1.2	0.9	0.4	15.2	25.8	23.8	14.2	3.5	3.3	3.4	3.7	4.3	5.0	2.2	0.5	0.4	0.5	0.3	1.1	1.2	1.5	1.7	1.8	1.8	1.6	1.3	1.1	1.3	1.2	1.1	1.0	0.8	0.7	0.4	0.2	0.1	0.4	0.7	0.5	0.3	0.2	0.1	0.2	0.2	0.2	0.2	0.2	0.1
3.0	23	1.1	1.3	1.4	1.4	1.2	0.8	10.7	20.9	22.7	16.8	7.7	7.3	7.2	7.3	7.6	8.1	5.0	2.8	1.2	0.4	0.2	1.0	1.2	1.5	1.7	1.8	1.7	1.5	1.2	1.0	1.1	1.1	1.0	0.9	0.7	0.6	0.4	0.2	0.1	0.3	0.6	0.4	0.2	0.1	0.1	0.2	0.2	0.2	0.2	0.2	0.1
2.8	22	1.0	1.3	1.4	1.4	1.2	0.9	6.2	13.0	16.4	14.7	9.2	8.5	8.1	7.9	8.0	8.1	7.1	5.5	3.5	1.6	0.2	0.8	1.1	1.4	1.6	1.7	1.6	1.4	1.1	0.8	0.9	0.8	0.8	0.7	0.6	0.5	0.3	0.2	0.1	0.2	0.4	0.3	0.1	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.1
1.9	21	0.8	1.0	1.1	1.0	0.9	0.7	2.3	4.8	6.6	7.0	5.6	4.7	4.4	4.3	4.4	4.6	7.7	8.4	6.6	3.5	0.1	0.6	1.1	1.4	1.6	1.7	1.6	1.4	1.0	0.6	0.5	0.5	0.5	0.5	0.5	0.4	0.3	0.2	0.1	0.1	0.2	0.1	0.1	0.2	0.2	0.3	0.3	0.2	0.2	0.1	0.0
1.5	20	0.5	0.5	0.5	0.5	0.3	0.1	1.3	1.8	1.5	1.2	1.6	1.5	2.6	3.8	4.9	6.0	6.6	11.5	10.6	5.9	0.0	0.7	1.2	1.5	1.7	1.7	1.6	1.4	1.1	0.6	0.0	0.1	0.2	0.3	0.3	0.3	0.3	0.2	0.2	0.1	0.0	0.1	0.2	0.3	0.3	0.3	0.3	0.3	0.2	0.1	0.0
1.7	19	0.8	0.8	0.8	0.7	0.5	0.2	1.6	2.4	2.6	2.1	1.6	2.2	3.0	3.7	4.4	5.1	4.7	8.6	8.3	4.8	0.5	1.7	2.2	2.6	2.7	2.8	2.6	2.3	1.9	1.4	0.8	0.7	0.6	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.6	0.4	0.3	0.2	0.2	0.2	0.1	0.1	0.1	0.2	0.2
1.9	18	0.9	1.0	1.0	0.9	0.7	0.4	1.7	2.7	3.1	3.1	2.6	2.9	3.2	3.4	3.8	4.1	3.0	5.7	5.7	3.4	0.6	2.7	3.2	3.5	3.7	3.7	3.5	3.1	2.7	2.1	1.4	1.3	1.2	1.1	1.0	1.0	0.9	0.9	0.9	0.9	1.0	0.8	0.6	0.5	0.5	0.4	0.4	0.4	0.3	0.4	0.4
2.0	17	0.8	1.0	1.1	1.0	0.8	0.5	1.7	2.5	3.0	3.2	3.0	2.9	2.9	2.8	2.8	2.9	1.7	3.2	3.3	1.9	0.6	3.6	4.0	4.2	4.4	4.4	4.2	3.8	3.2	2.6	1.9	1.7	1.6	1.5	1.4	1.3	1.3	1.3	1.2	1.2	1.2	1.1	0.9	0.8	0.7	0.6	0.6	0.5	0.5	0.5	0.5
1.8	16	0.3	0.8	1.1	1.1	0.9	0.6	1.3	1.9	2.2	2.3	2.1	2.1	2.0	1.7	1.5	1.5	0.9	1.2	1.3	0.8	0.3	4.2	4.5	4.8	4.9	4.9	4.6	4.2	3.6	2.8	2.1	2.0	1.8	1.7	1.6	1.6	1.5	1.5	1.4	1.4	1.4	1.2	1.1	1.0	0.9	0.8	0.7	0.7	0.6	0.6	0.6
1.5	15	0.6	0.4	0.9	1.0	0.9	0.6	0.8	0.8	0.6	0.3	0.5	0.3	0.3	0.2	0.2	0.3	0.2	0.4	0.2	0.2	0.6	4.7	4.9	5.1	5.2	5.1	4.9	4.4	3.8	3.0	2.2	2.1	1.9	1.8	1.7	1.6	1.6	1.5	1.5	1.4	1.4	1.3	1.1	1.0	0.9	0.8	0.8	0.7	0.7	0.6	0.6
2.5	14	1.8	3.1	4.2	5.0	5.4	5.7	5.2	4.4	3.5	2.5	1.4	0.7	0.5	0.7	1.4	2.7	3.2	3.2	2.5	1.2	0.5	4.9	5.1	5.3	5.3	5.2	4.9	4.5	3.8	2.9	2.1	2.0	1.9	1.8	1.7	1.6	1.5	1.5	1.4	1.4	1.4	1.2	1.1	1.0	0.9	0.9	0.8	0.7	0.7	0.6	0.5
3.3	13	6.7	7.7	8.4	8.9	9.3	9.6	8.5	7.2	5.7	4.2	2.6	0.7	0.5	0.7	2.1	4.6	5.4	5.3	4.3	2.4	0.4	4.8	5.1	5.2	5.2	5.0	4.7	4.3	3.6	2.7	1.8	1.7	1.6	1.6	1.5	1.4	1.4	1.3	1.3	1.2	1.2	1.1	1.0	0.9	0.8	0.8	0.7	0.7	0.6	0.5	0.5
3.5	12	9.7	9.8	9.9	10.0	10.1	10.2	9.1	7.8	6.3	4.6	2.9	0.7	1.7	1.1	1.4	5.0	6.4	6.6	5.5	3.2	0.3	4.3	4.8	4.9	4.9	4.7	4.4	3.9	3.3	2.3	1.4	1.3	1.3	1.2	1.2	1.1	1.1	1.0	1.0	0.9	0.9	0.8	0.8	0.7	0.7	0.7	0.6	0.6	0.5	0.4	0.4
2.8	11	6.9	6.9	6.8	6.8	6.8	6.6	6.5	5.8	4.8	3.6	2.1	2.3	3.9	3.3	0.9	3.5	6.1	6.9	6.2	3.8	0.3	3.5	4.3	4.5	4.5	4.3	4.0	3.5	2.8	1.8	0.8	0.8	0.8	0.7	0.7	0.7	0.7	0.6	0.6	0.6	0.5	0.5	0.5	0.5	0.5	0.5	0.4	0.4	0.3	0.3	0.2
1.6	10	0.1	0.2	0.3	0.1	0.2	0.5	0.6	1.2	1.3	0.9	0.2	5.1	7.1	6.7	4.3	0.2	4.4	6.3	6.2	4.0	0.3	2.4	3.5	3.9	3.9	3.8	3.5	3.0	2.3	1.4	0.0	0.1	0.1	0.2	0.2	0.2	0.2	0.1	0.1	0.1	0.0	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.1	0.0
1.9	9	3.0	1.9	1.0	0.3	0.3	0.8	1.2	1.5	1.7	1.7	1.7	3.0	4.6	5.8	6.4	6.6	6.1	5.2	3.7	1.8	0.3	2.4	3.4	3.7	3.6	3.3	3.0	2.5	1.9	1.1	0.8	0.8	0.7	0.7	0.7	0.7	0.7	0.8	0.8	0.8	0.9	0.7	0.6	0.5	0.4	0.3	0.3	0.2	0.2	0.2	0.2
2.2	8	5.4	3.9	2.7	1.5	0.5	0.3	1.1	1.7	2.2	2.6	3.0	3.4	4.7	5.7	6.2	6.2	5.7	4.7	3.3	1.5	0.4	2.4	3.3	3.4	3.3	2.9	2.6	2.1	1.6	1.2	1.5	1.4	1.4	1.4	1.4	1.4	1.4	1.5	1.5	1.5	1.6	1.4	1.2	1.0	0.9	0.7	0.6	0.5	0.4	0.4	0.4
2.5	7	7.2	5.5	3.9	2.5	1.2	0.2	0.9	1.8	2.6	3.2	3.8	3.6	4.8	5.5	5.9	5.8	5.2	4.3	2.9	1.3	0.5	2.4	3.1	3.2	3.0	2.6	2.2	1.8	1.5	1.4	1.9	1.9	1.9	1.8	1.9	1.9	1.9	2.0	2.0	2.0	2.1	1.8	1.6	1.4	1.2	1.0	0.9	0.7	0.6	0.5	0.5
2.6	6	8.5	6.6	4.8	3.2	1.8	0.6	1.0	1.9	2.8	3.6	4.4	3.8	4.7	5.3	5.5	5.4	4.8	3.8	2.6	1.1	0.5	2.3	3.0	3.0	2.7	2.3	1.9	1.6	1.5	1.6	2.2	2.2	2.1	2.1	2.1	2.2	2.2	2.2	2.3	2.4	2.4	2.1	1.9	1.6	1.4	1.2	1.1	0.9	0.8	0.6	0.6
2.6	5	9.1	7.1	5.3	3.6	2.1	0.8	0.8	1.9	2.9	3.7	4.5	3.7	4.5	5.0	5.2	4.9	4.4	3.5	2.3	0.9	0.5	2.2	2.8	2.8	2.5	2.1	1.7	1.4	1.4	1.7	2.3	2.3	2.2	2.2	2.2	2.3	2.3	2.3	2.4	2.5	2.5	2.2	2.0	1.7	1.5	1.3	1.1	1.0	0.8	0.7	0.6
2.5	4	9.0	7.1	5.3	3.7	2.2	0.8	0.6	1.7	2.7	3.5	4.3	3.5	4.2	4.7	4.7	4.5	3.9	3.1	2.0	0.8	0.5	2.0	2.7	2.7	2.3	1.9	1.5	1.3	1.4	1.7	2.3	2.2	2.1	2.1	2.1	2.1	2.2	2.2	2.3	2.4	2.4	2.2	1.9	1.7	1.5	1.3	1.1	0.9	0.8	0.7	0.6
2.2	3	8.1	6.4	4.8	3.4	2.0	0.8	0.6	1.5	2.4	3.2	3.9	3.2	3.8	4.2	4.3	4.0	3.5	2.7	1.8	0.7	0.4	1.9	2.5	2.5	2.2	1.8	1.4	1.2	1.2	1.5	2.0	1.9	1.8	1.8	1.8	1.8	1.9	1.9	2.0	2.1	2.2	1.9	1.7	1.5	1.3	1.1	1.0	0.8	0.7	0.6	0.5
1.8	2	6.4	5.1	3.8	2.6	1.5	0.5	0.5	1.3	2.0	2.6	3.1	2.6	3.3	3.7	3.7	3.5	3.1	2.4	1.6	0.7	0.3	1.7	2.3	2.4	2.1	1.7	1.3	1.1	1.0	1.1	1.5	1.4	1.3	1.3	1.3	1.3	1.3	1.4	1.5	1.6	1.7	1.4	1.3	1.1	1.0	0.9	0.7	0.6	0.5	0.5	0.4
1.3	1	3.8	2.9	2.1	1.3	0.7	0.4	0.6	1.0	1.3	1.6	1.8	2.0	2.6	3.0	3.2	3.0	2.6	2.1	1.4	0.6	0.2	1.5	2.2	2.3	2.1	1.8	1.4	1.1	0.9	0.7	0.8	0.7	0.6	0.6	0.5	0.5	0.6	0.6	0.7	0.8	0.9	0.8	0.7	0.6	0.5	0.5	0.4	0.3	0.3	0.2	0.2
0.8	0	0.2	0.4	0.7	1.0	1.2	1.3	1.3	1.1	0.9	0.5	0.1	1.1	1.9	2.4	2.5	2.5	2.2	1.8	1.2	0.6	0.0	1.4	2.0	2.2	2.1	1.8	1.5	1.2	0.9	0.5	0.0	0.2	0.3	0.4	0.4	0.4	0.4	0.4	0.3	0.2	0.0	0.1	0.1	0.2	0.2	0.2	0.1	0.1	0.1	0.1	0.0

Average CPU time even over the hardest and most error prone region ($North \in \{20,30\}, East \in \{0,10\}$) is 14.7211 seconds with an average angular error of 9.5877 degrees. The final improvement of the varied mesh library on interpolation results is clearly evident in Figure 15. The actual error values are listed in Table 6, as well as the average error along each x and y value, clearly showing that the average final heading error averaged along one direction is reduced to a maximum of 5.0 degrees.

As a final test, we computed optimal trajectories using the PS method at the midpoints of the database library, five meters in between known optimal trajectories in the North and East directions, and compared the results to interpolated trajectories at the same locations. This five by five test target region represents the most challenging locations for both the PS and the interpolation methods to solve because each point is the maximum distance from the nearest know optimal trajectory in the library. The summarization of the results comparing the optimal trajectories solved using the pseudospectral method compared to the two-dimension interpolations are listed in Table 7, and the actual results are listed in Table 8.

Table 7. Summarization of pseudospectral vs interpolation results for test targets

Method	mean CPU time	Mean Final Error	Mean t_f
Pseudospectral	265.93 sec	0.2827 degrees	2.73 sec
Interpolation	7.10 sec	9.1137 degrees	3.41 sec

Table 8. Pseudospectral vs. 2-D interpolation method results over *North*, *East* ∈ {5, 10, …45}

PS Alpha Minimum Time Results (in seconds)						
Target Position		North				
		5	15	25	35	45
East	5	6.553	6.715	2.845	2.470	1.119
	15	5.340	5.227	2.811	2.259	0.872
	25	3.596	2.732	2.289	1.312	0.931
	35	3.137	2.484	1.914	1.513	1.216
	45	3.024	2.538	2.090	1.740	1.460

PS Alpha CPU Time Results (in seconds)						
Target Position		North				
		5	15	25	35	45
East	5	121	176	70	83	185
	15	313	75	2834	271	205
	25	67	71	126	155	290
	35	78	68	42	211	489
	45	104	127	81	288	118

2D Interpolation Minimum Time Results (sec)						
Target Position		North				
		5	15	25	35	45
East	5	6.595	7.205	6.789	3.270	2.041
	15	5.735	6.077	5.000	2.669	1.787
	25	4.413	4.673	2.788	2.251	1.105
	35	3.527	2.791	2.192	1.551	1.202
	45	3.306	2.755	2.222	1.804	1.479

2D Interpolation CPU Time Results (sec)						
Target Position		North				
		5	15	25	35	45
East	5	11.437	10.853	6.788	8.879	6.938
	15	9.694	10.197	10.191	10.371	6.975
	25	8.324	4.539	6.251	7.760	4.709
	35	6.174	5.454	4.428	5.636	5.481
	45	6.738	5.804	5.083	3.519	5.304

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

Progress was made toward demonstrating a faster and less computationally expensive method of finding optimal trajectories. Generating the library of optimal trajectories proves the pseudospectral method is extremely useful in computing optimal trajectories, and comparing the trajectories flown by the HUAV based on the mathematical models to recommended maneuvers from the "The Army Aviator's Handbook for Maneuvering Flight and Power Management" validates the algorithm's results.

Most notable is the use of the "Pitch-back turn" for targets within twenty meters from the nose of the HUAV's initial state. These are extremely similar to the trajectory flown by US Army AH-64 D longbow pilots, in order to re-orient to the target in minimal time and suppress with organic munitions. At such close ranges, the helicopter cannot simply turn directly toward the target. The helicopter pilot or AUV must take into consideration forward airspeed, altitude, and maneuvering operational limits. With targets in such close proximity, the minimal time trajectory actually requires the aircraft to maneuver away from the target while trading airspeed for altitude, then stabilize at a final heading towards the target, much like seen early in Figure 5.

While the PS method was proven accurate, it is also time consuming. It could and should be utilized for solving optimal trajectories offline, but would not be useful in real-time as an online solver for a HUAV flying

autonomously in the present configuration. This is where the interpolation method proved its worth. Although not as accurate, it did provide approximate solutions with acceptable error for the majority of targets. In regions where the bifurcation of optimal trajectory solutions is present, a smaller mesh library is needed and was used to bring error down to acceptable results.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

1. Incorporate Interpolation Method in Three Dimensions

Although positive results were obtained using a two dimensional interpolation method, HUAVs operate in three dimensional space, so a three dimensional interpolation method is required to fully make the method viable for operational use. Therefore, to truly validate the interpolation method, we would incorporate targets that vary in all three dimensions, as well as varying the initial state of the HUAV specifically in altitude and velocity to ensure the method is both robust and stable.

2. Further Investigation into Variable Mesh

I believe that even further reduced error could be achieved if an adaptive grid size was utilized. This adaptive grid would use larger intervals in regions where optimal trajectories are very similar and smaller intervals in the regions with angular error greater than the desired threshold. If this adaptive mesh library is populated offline prior to utilization, the overall interpolation method should see an even larger computation time reduction.

3. Code for an Onboard Platform

Additionally, if the PS method code and the interpolation routines created for this thesis were converted into C++ code that is device-independent, the method could possible by used in real-time on an actual HUAVs. C++ or a similar high level computer language should reduce the compilation and run time compared to the MATLAB code which is FORTRAN based. Additionally, if the code were loaded onto a hardware platform designed specifically for the trajectory pseudospectral and interpolation algorithms, further speed improvements could be realized.

4. Incorporate a Closed-Loop Feedback Control System

Finally, to ensure the interpolated control inputs meet the performance constraints and bounds, a closed-loop feedback system could be incorporated that continually updates the control inputs provided to the aircraft based on the current states of the system. This would help ensure that no constraints are violated as well as bringing the final state of the aircraft to the intended point. It was noted while evaluating the two dimension interpolation results that often the trajectory provided actually brought the heading of the UAV directly onto the target, but the control inputs continued maneuvering the HUAV away from the target, based on the assumption of the linear combination of the final time.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] M. E. Dempsey, "Eyes of the Army – U.S. Army Roadmap for Unmanned Aircraft Systems 2010–2035," U.S. Army UAS Center of Excellence, Ft. Rucker, Alabma, 9 April 2010.

[2] C. Thompson, "Critical and Essential- UAV as a Wingman," Apache News, Boeing.com, [Online]. Available: http://www.boeing.com/apachenews/2009/issue_01/army_s2_p2.html. [Accessed 14 May 2012].

[3] D. J. Snyder, "Design Requirements For Weaponizing Man-Portable Uas In Support Of Counter-Sniper Operations," M.S. Thesis, Naval Postgrad., Monterey, CA, September 2011.

[4] R. M. Murray *et al.*, "A Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems," in *Conf. on Decision and Control*, 2000.

[5] C. Atkeson, "Using Trajctory Optimizers to Speed Up Global Optimization in Dynamic Programming," in *Neural Information Processing Systems*, Morgan Kaufmann, 1994, p. Chapter 6.

[6] C. G. Atkeson and L. Chenggang, "Standing Balance Control Using a Trajectory Library," in *IEEE/RSJ Conference on Intelligent Robots and Systems*, St Louis, 2009.

[7] R. M. Murray *et al.*, "Nonlinear Trajectory Generation for Unmanned Air Vehicles with Multiple Radars," in *IEEE Conf. on Decision and Control*, 2004.

[8] M. B. Milam, "Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems," PhD thesis, Control and Dynamical Systems, Caltech, 2003.

[9] C. P. Neuman and A. Sen, "A Suboptimal Control Algorithm for Constrained Problems using Cubic Splines," *Automatica,* vol. 9, pp. 601-613, 1973.

[10] J. Y. S. Luh *et al.*, "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots,," *IEEE Trans. Autom. Control,* Vols. AC-28, no. 12, pp. 1066-1074, 1983.

[11] B. T. Gatzke, "Trajectory Optimization for Helicopter Unmanned Aerial Vehicles (UAVs)," M.S. Thesis, Dept. Math., Naval Postgrad., Monterey, CA, 2010.

[12] F. Fahroo and I. M. Ross, "Costate Estimation by a

Legendre Pseudospectral Method," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Boston, MA, 10-12 August 1998.

[13] W. Kang *et al.*, "A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems," *IEEE Trans. Automat. Contr.,* vol. 51, no. 7, pp. 1115-1129, 2006.

[14] W. Kang *et. al*, "Minimum time control of helicopter UAVs using computational dynamic optimization," in *Proceedings of IEEE Conf. on Control and Automation*, Santiago, Chile, December 2011.

[15] W. Kang *et al.*, "On the Convergence of Nonlinear Optimal Control Using Pseudospectral Methods for Feedback Linearizable Systems," *International Journal of Robust and Nonlinear Control,* vol. 17, pp. 1251-1277, 2007.

[16] G. Kai *et al.*, "Design and Implementation of a Robust and Nonlinear Flight Control System for an Unmanned Helicopter," National University of Singapore.

[17] G. Cai *et al.*, *Unmanned Rotorcraft Systems*, New York: Springer, 2011.

[18] U.S. Army Aviation Center and School, *FM 3.140 Helicopter Gunnery*, Ft. Rucker, AL: Department of the Army, 06/16/2006.

[19] P. Gill *et al.*, "Snopt: An SQP algorithm for large-scale constrained optimization," *SIAM Journal on Optimization,* vol. 12, no. 4, p. 979-1006, 2002.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Wei Kang
   Naval Postgraduate School
   Monterey, California

4. Hong Zhou
   Naval Postgraduate School
   Monterey, California

5. Carlos Borges
   Naval Postgraduate School
   Monterey, California